Ground Segment Systems Engineering Handbook

The Aerospace Corporation
Ground Segment Systems Engineering Handbook
Technical Operating Report TOR-2016-01797

AEROSPACE REPORT NO.
TOR-2016-01797

GROUND SEGMENT SYSTEMS ENGINEERING HANDBOOK

Edited by

GAIL A. JOHNSON-ROTH, GERALDINE A. CHAUDHRI,
and WILLIAM F. TOSNEY

August 1, 2016

Systems Planning, Engineering, and Quality
THE AEROSPACE CORPORATION
El Segundo, CA 90245-4691

Introduction

Gail Johnson-Roth
Enterprise Systems Engineering
Corporate Chief Engineer's Office
Geraldine Chaudhri
Software Systems Assessment
Software Systems Assurance Department

The importance of the ground segment in assuring space system mission success cannot be overstated. In many cases, the mission cannot be accomplished without the logic, data, and processing that the ground segment provides. The Global Positioning System (GPS) payload, for example, cannot provide its functionality without the navigation solution that the ground segment calculates and uploads. Most military communication satellite systems rely completely on the ground segment to perform the difficult job of scheduling communication resources.

It is these critical roles that have prompted the need for this *Ground Segment Systems Engineering Handbook*. It identifies and describes the systems engineering functions required of a government program office in the acquisition of the ground segment (though many of the functions described are also directly applicable to contractor organizations). This handbook can be used as a training manual, mentoring tool, or refresher for the novice or intermediate systems engineer tasked with helping to acquire, develop, and operate a ground segment for a space system.

The generic guidance provided here is not intended to address every unique requirement of every ground segment, but presents a consensus opinion and overview. It draws upon a reference architecture (Figure 1) that was developed as a touchstone to help focus and organize the content of the book. Chapters are dedicated to each of the elements and functions in the reference architecture as well as to their supporting processes and disciplines.

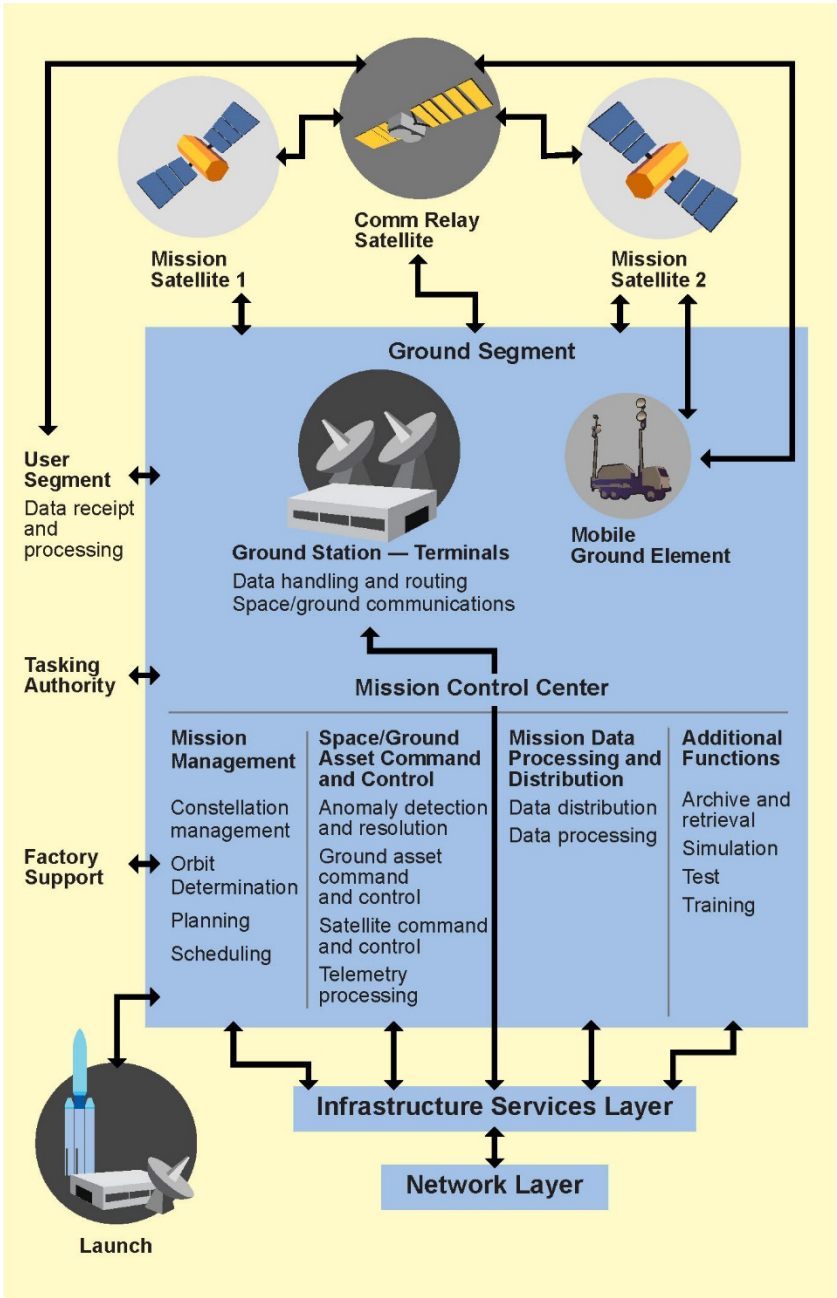


Figure 1. Representative ground segment reference architecture with external interfaces.

The handbook is organized as follows:

- Ground Segment Elements. Chapters 1–9 provide an overview of the ground segment based on the reference architecture (Figure 1). Individual chapters further elaborate on each of the significant elements, functions, and infrastructure services, including ground station terminals, the mission control center, mission management, space/ground asset command and control, data processing and distribution, and operations and training.

- Engineering Processes and Products. Chapters 10–13 provide details on system engineering, software, hardware, and facilities. These functions and products provide the underlying framework for process and product definition and development at the earliest phases of the acquisition cycle. The design of the ground segment hardware is complex, with built-in redundancy, and the software is composed of thousands of lines of code developed over long periods of time. A successful space mission can only be accomplished with effective systems engineering processes to produce the fully integrated ground segment.

- Acquisition and Operations. Chapters 14–29 provide the instructional material to guide the systems engineer through each phase of the ground segment development cycle, from concept development through sustainment. A special chapter on technology refresh recognizes the longevity of legacy systems and the need to address obsolescence while maintaining capability. There are also a number of program management and system engineering chapters (risk management, configuration management, software metrics, quality assurance, readiness reviews) that highlight the need for active engagement between the system engineering and program management teams to ensure an optimal architecture is identified and developed in the most cost-effective manner possible, considering performance, cost, and risk.

- Supporting Disciplines. Chapters 30–34 provide detailed material on some of the most important tasks required to implement each of the systems engineering disciplines that span the ground segment development cycle (reliability, availability, and maintainability [RAM]; fault management; cybersecurity; human systems integration; system safety). An effective approach to reliability and maintainability can help prevent failures that may result in catastrophic anomalies or degradation of the space system assets (both ground and space). Existing standards are mostly silent on availability as a performance

measure. The information provided in these chapters gives an overview of the RAM tasks, as wells as human factors and system safety, to be accomplished by ground systems over a program lifecycle to support the success of a space mission.

The majority of the ground segments today may represent “behind the times” technology; however, these legacy systems have functioned for a long time and will continue to operate for years to come. The *Ground Segment Systems Engineering Handbook* was designed to the capture the current state of affairs. Newer technologies will push ground systems toward a service-based architecture that offers greater agility and intersects with strategic partners. In the meantime, however, this handbook will provide the guidance and roadmap needed to understand the details of the system engineering functions and their practical applications, methodologies, evaluation criteria, products, and processes throughout the development cycle as applied to systems that exist today.

Foreword

William F. Tosney
Corporate Chief Engineering
Systems Planning, Engineering, and Quality

In spring of 2015, the Government Accountability Office delivered its annual report to Congress on DOD space acquisitions. The report cited stunning cost increases for several major programs, including some high-profile ground systems. The report also noted that in some cases, the space segment was effectively waiting for the ground segment to come on line. When satellites are placed on orbit without corresponding ground systems in place, the report said, their capability is effectively wasted, as a portion of their limited lifespan is spent without being fully utilized.

Recently, the Air Force Space and Missile Systems Center commissioned a series of studies from The Aerospace Corporation to investigate best practices and driving trends in the industry. These studies were intended to develop general guidelines that would lead to more affordable and resilient ground systems while preventing some of the delays and cost overruns of the past. Much of that knowledge has been collected in this *Ground Segment Systems Engineering Handbook*.

In compiling this manual, we sought to include solid guidance based on the existing state of ground systems. As for the future, that remains a moving target. Ground systems typically evolve incrementally, because—unlike spacecraft—they can be upgraded, even after they've been placed in service. At Aerospace, we feel that this mentality—the belief that "we can always fix it later"—does not really serve the best interests of the ground segment community. Instead, we believe that ground systems must be fully engineered and tested to meet a verifiable list of user requirements—just like launch and space vehicles. That is the philosophy that underlies this book.

Although it may be hard to paint a clear picture of future ground segments, a few new considerations are certain to play a role. Budget constraints, both in acquisitions and operations, will motivate designers to develop new cost-effective approaches to designing, acquiring, and sustaining ground systems. This will foster an embrace of cloud computing—a model for ubiquitous, on-demand access to a shared pool of configurable computing resources that can be rapidly provisioned and readily scaled. New emphasis on the enterprise architecture, big-data and data analytics, resiliency, cyber security, and model-based engineering will contribute to fundamentally different approaches to systems engineering along with requisite design and verification processes.

Already, a greater emphasis is being placed on increased automation and reduced staffing on the operations side. Future ground systems will still need to accommodate legacy components while being expandable, adaptable, and resilient enough to address unforeseen requirements.

Cloud computing will enable the adoption of service-oriented architectures, which provide a common set of services with similar user interfaces. For example, with Software as a Service, users do not purchase software, per se, but rather license the software functions on a subscription basis; anyone who uses web-based email will be familiar with this paradigm. With Platform as a Service, users can develop and host their own programs on the provider's servers. With Infrastructure as a Service, users essentially rent the provider's network and administrative functions, which can be scaled up or down as requirements change.

Any of these models could be achieved through a layered architecture based on the commercially successful service-provider model, which is executed in a cloud-based environment using commodity processors. This approach enables the migration of existing capabilities, without the disruption to users, as well as the deployment of future capabilities in a faster and more economical manner.

Cyber security, of course, will also impose new challenges and will need to be addressed from the earliest design phases to ensure a failsafe architecture. Cyber security is a facet of overall ground system resiliency, or the ability to support the functions necessary for mission success in spite of hostile action or adverse conditions. Achieving resiliency in a service-provider environment will be no small challenge.

The next-generation ground segment will achieve greater flexibility and economy through the use of common applications, common infrastructure, structured software development, cyber resilience, and automation. At the same time, legacy systems will be modernized and transformed to serve an expanded set of mission requirements. As such, they may be subject to a different outlook with regard to how they are designed, developed, test, fielded, operated, and maintained. Nonetheless, they will still benefit from the rigorous and comprehensive systems engineering approach that is promoted in this guide.

This handbook presents best practices as we know them today, based on years of experience and coordinated effort. It is provided for the benefit not only members of The Aerospace Corporation, but also government program offices and contractor personnel. We hope you will find it useful in developing the ground systems of tomorrow.

Preface

Specifications, Standards,
Policies, Handbooks, and Best Practices

Gail A. Johnson-Roth
Enterprise Systems Engineering Office
Corporate Chief Engineer's Office
Brian E. Shaw
Engineering and Integration Division
Space Systems Group

There are current standards, handbooks, and guides for a disciplined systems engineering (SE) and mission assurance (MA) approach applicable to the development of national security space ground segments/systems. These documents were developed, for the most part, in collaboration among government, industry, and academic subject matter experts who strive to address high-priority common challenges facing the space community and the challenge to achieve 100-percent mission success.

Specifications and Standards

Specifications and standards are commonly used as contractual compliance documents and should be evaluated for customer/agency and program applicability, tailored as necessary, and implemented as contractually compliant requirements. The list is current as of this document's publication date and has a stable heritage, but may evolve as standards are updated or as program needs/experience evolve. The Aerospace Corporation (Aerospace) recommended specifications and standards (identified in the list with an asterisk) are those standards deemed necessary to adequately support and guide the successful implementation of proven engineering and program management practices in US space programs to achieve mission success while at the same time minimizing unwarranted and costly impacts to system performance and program schedule. Aerospace's government customers have compliance lists or other organizational governance for usage of standards that is consistent with the Aerospace recommended standards but tailored in scope for specific product domains or organizational practices.

Specific emphasis is placed on current authoritative documents that are applicable to the customers of Aerospace, but the list should not be considered all-inclusive. Other government and commercial documents, and contractor proprietary processes or command media, for ground systems may exist but not evaluated or documented in this report.

Program Management

*Schipper, Gary A. *Program and Subcontractor Management*. TR-RS-2008-00019, AKA[1] TOR-2008(8583)-7731. The Aerospace Corporation, El Segundo, CA. March 2008. [also published as[2] SMC-S-019, Rev A. 2008]

IEEE Standards Association. IEEE 15288.2. *IEEE Standard for Technical Reviews and Audits on Defense Programs*. The Institute of Electrical and Electronics Engineers, Inc., New York, NY. May 15, 2015.

*SAE International. EIA649-1. Configuration Management Requirements for Defense Contracts. Warrendale, PA. 2014.

*Shaw, Brian E. *Tailoring of EIA-649-1: Definition of Major (Class I) Engineering Change Proposal*. TOR-2015-01904. 2014, [also published as SMC-T-007. 2015], The Aerospace Corporation, El Segundo, CA. 2015.

*SAE International. AS 6500. *Manufacturing Management Program*. Warrendale, PA. 2014.

*International Organization for Standards. ISO 17666. *Space Systems – Risk Management*. Switzerland. 2003.

Donahue, Charles P. and B. McKinzey. *Configuration Management*. TR-RS-2008-00002, The Aerospace Corporation, El Segundo, CA. 2005.

ANSI/EIA 748C. *Earned Value Management System*. Arlington, VA. 2014.

MIL-HDBK-881A, *Work Breakdown Structures for Defense Materiel Items,* Department of Defense Handbook, Washington D. C., DOD, 2005.

*Perestegy, L. B. and C. E. O'Conner. *Technical Reviews and Audits for Systems, Equipment, and Computer Software,* TR-RS-2009-00021. The Aerospace Corporation, El Segundo, CA.

[1] AKA – Also Known As. Technical reports of The Aerospace Corporation published prior to 2014 that are used as standards were renumbered from the original TOR report number to a TR-RS (Technical Report – Recommended Standard) nomenclature. Reports under both TOR and TR-RS nomenclatures contain identical content.

[2] Technical reports of The Aerospace Corporation that are used as standards are also published as USAF Space and Missile Systems Center (SMC) standards with a government approval page. Reports under both Aerospace and SMC nomenclatures contain identical content.

Systems Engineering

*ISO/IEC/IEEE 15288. Systems and Software Engineering – System lifecycle processes. 2015.

*ISO/IEC/IEEE Computer Society. IEEE 15288.1. IEEE Standard for Application of Systems Engineering on Defense Programs. Washington, D.C. 2014.

*Shaw, Brian E. *Tailoring of: IEEE-15288.1 Specialty Engineering Supplement.* TOR-2015-01949. [also published as SMC-T-006. 2015] The Aerospace Corporation, El Segundo, CA. 2015.

*American Institute of Aeronautics and Astronautics. AIAA-S-117-2010. *Space System Verification Program and Management Process*. Reston, WA. 2010.

*Shaw, Brian E. *Systems Engineering Requirements and Products*. TR-RS-2013-00001, The Aerospace Corporation, El Segundo, CA. 2013.

SAE International. ANSI/EIA 632, *Processes for Engineering a System*. Warrendale, PA. 2003.

Institute of Electrical and Electronics Engineers. IEEE 1220, *IEEE Standard for Application and Management of the Systems Engineering Process*. Washington, DC. 2005.

Electronic Industries Alliance. EIA-731. *Systems Engineering Capability Model*. Arlington, VA. 2002.

Product Assurance

*SAE International. AS9100, Rev. C. Quality Systems – Aerospace – Model for Quality Assurance in Design, Development, Production, Installation, and Servicing. Warrendale, PA. 2009.

Richter, Eric S. TR-2014-01196, *Quality Space and Launch Requirements Addendum to AS9100C TR-RS-2014-00003*, [also published as SMC-S-003. 2015], The Aerospace Corporation, El Segundo, CA. 2015.

Program Protection

*DOD 5220-22M. National Industrial Security Program. 2006.

*DODI 8500.2. Information Assurance Implementation. 2003.

*DODI 5200.39, incorporating change 1. Critical Program Information (CPI) Protection within DOD. 2015.

DODI 8510.01. Risk Management Framework (RMF) for DOD Information Technology (IT). 2014.

Committee on National Security Systems Instruction (CNSSI) 1253. *Security Categorization and Control Selection for National Security Systems*. Fort Meade, MD. 2012.

Office of the Director of National Intelligence. Intelligence Community Directive Number 503. Intelligence Community Information Technology System Security Risk Management, Certification, & Accreditation. 2008.

Air Force Pamphlet 63-113. Program Protection Planning for Life Cycle Management. 2013.

Air Force Policy Directive 63-17. Technology and Acquisition Systems Security Program Protection. 2001.

Electrical Power

*Dunbar, Mark W. *Electromagnetic Compatibility Requirements for Space Equipment and Systems*. TR-RS-2008-00008, AKA TOR-2005(8583)-1, Rev A, [also published as SMC-S-008. 2009], The Aerospace Corporation, El Segundo, CA. 2008.

*MIL-STD-461. Requirements for the Control of Electromagnetic Interference Characteristics of Subsystems and Equipment. 2015.

*MIL-STD-1542B. Electromagnetic Compatability Grounding Requirements for Space System Facilities. 1991.

Parts, Materials, and Processes

*MIL-STD-3018. *Parts Management*. 2015.

*Robertson, Steven R., Lawrence I. Harzstark, Joyce P. Siplon, David M. Peters, Paul H. Hesse, Michael J. Engler, George G. Cuevas, and David C. Meshel. *Parts, Materials and Processes Control Program for Space and Launch Vehicles*. TR-RS-2013-00024, TOR-2006(8583)-5235 Rev B, [also published as SMC-S-009. 2013], The Aerospace Corporation, El Segundo, CA. 2008.

*Robertson, Steven R., Lawrence I. Harzstark, Joycelyn P. Siplon, David M. Peters, Paul H. Hesse, Michael J. Engler, Robert J. Ferro, Wayne A. Martin, George G. Cuevas, Melvin H. Cohen, Jeffrey H. Sokol, and Gary J. Ewell. *Technical Requirements for Electronic Parts, Materials & Processes Used in Space and Launch Vehicles* TR-RS-2013-00010, TOR-2006(8583)-5236 Rev B, [also published as SMC-S-010. 2013], The Aerospace Corporation, El Segundo, CA. 2008.

Software

*Adams, Richard J., Suellen Eslinger, Karen L. Owens, and Joanne M. Tagami. *Software Development Standard for Mission Critical Systems*. TR-RS-2015-00012, [also published as SMC-S-012. 2015.] The Aerospace Corporation, El Segundo, CA. 2014.

*ISO/IEC 15939. Systems and Software engineering –Measurement Process. 2007.

IEEE/EIA 12207-2008. Systems and Software Engineering: Software Life Cycle Processes

Interoperability

*DOD Arch V2.02. DOD Architecture Framework (DODAF).2010.

*Department of Defense, Defense Information Technology Standards Registry (DISR). Current version. https://gtg.csd.disa.mil

Reliability and Maintainability

*Ingram-Cotton, John B., Myron J. Hecht, Roland J. Duphily. *Reliability Program Requirements for Space Systems*. TR-RS-2007-00013, TR-2007(8583)-6689, [also published as SMC-S-013. 2008], The Aerospace Corporation, El Segundo, CA. 2007.

*MIL-STD-785B including Notices 1 and 2. Reliability Program for Systems and Equipment Development and Production. 1988.

MIL-STD-1629A. Procedures for Performing a Failure Mode, Effects and Critical Analysis. 1980.

*MIL-STD-470B. Maintainability Program for Systems and Equipment. 1989.

Survivability

*Cuevas, George G., John R. Wiggins. *Survivability Program Management for Space*. 2010. TR-RS-2010-00014 AKA TOR-2008(8583)-8164, Rev A, [also published as SMC-S-014], The Aerospace Corporation, El Segundo, CA. 2010.

Human Systems Integration

MIL-STD-46855. Human Engineering Requirements for Systems, Equipment, and Facilities. 2016.

*MIL-STD-1472G. Department of Defense Design Criteria Standard. Human Engineering. 2012.

*ANSI-HFES 100-2007. *Human Factors Engineering of Computer Workstations*. Santa Monica, CA: Human Factors and Ergonomics Society. 2007.

*Shaw, Brian E. *Human Computer Interface (HCI) Design Criteria Volume 1: User Interface Requirements*. [also published as SMC-S-023, Vol 1]. TR-RS-2009-00023 Vol1 AKA TOR-2010(8591)-2. The Aerospace Corporation, El Segundo, CA. 2009.

*Shaw, Brian E. *Human Computer Interface (HCI) Design Criteria Volume 2: Display Conventions for Space System Operations*. 2009. [also published as SMC-S-023, Vol 2.] TR-RS-2009-00023 Vol 2 AKA TOR-2010(8591)-3. The Aerospace Corporation, El Segundo, CA. 2007.

Integrated Logistics Support

*MIL-PRF-49506. Performance Specification: Logistics Management Information. 1996.

MIL-STD-1545. Optional Spare Parts, Maintenance and Inventory Support of Space and Missile Systems. 1977.

MIL-STD-1538. Spare Parts and Maintenance Support of Space and Missile Systems Undergoing RDT&E. 1973.

MIL-STD-130N. Identification Marking of US Military Property. 2007.

*MIL-STD-1367A, without Notice 1. Packaging, Handling, Storage, and Transportability Requirements for Systems and Equipments. 1989.

MIL-STD-1366E. Interface Standard for Transportability Criteria. 2006.

MIL-STD-2073/1E. Standard Practice for Military Packaging. 2008.

TMCR-86-01P. Air Force Technical Manual Contract Requirements (TMCR). 2014.

*MIL-PRF-29612B, including Notice 2. Training Data Products. 2001.

System Safety

*MIL-STD-882E. *System Safety*. 2012

*AFSPCMAN 91-710. *Range Safety User Requirements Manual*. Volumes 1–7. 2004.

Environmental

*NAS 411. Hazardous Materials Management Program. 2013.

*Chao, Chia-Chun, William S. Campbell, Glenn E. Petersen, and William H. Ailor. *Requirements of End-of-Life Disposal of Satellites Operating at Geosynchronous Altitude*. [Also published as SMC-S-015], TR-RS-2009-00015, AKA TOR-2006(8583)-4474, Rev A. The Aerospace Corporation, El Segundo, CA. 2009.

*Chao, Chia-Chun, William S. Campbell. *Requirements of End-of-Life Disposal of Satellites Operating in Orbits with Perigees below 2000 Kilometers*. [Also published as SMC-S-022], TR-RS-2007-00022, AKA TOR-2007(8506)-7154. The Aerospace Corporation, El Segundo, CA. 2009.

Test: Ground System

*Lutton, David, Colleen M. Ellis, James A. Shneer, Suellen Esli8nger, and Brian E. Shaw. *Test Requirements for Ground Systems*. [also published as SMC-S-024. 2013], TR-RS-2013-00024, AKA TR-2013-00215. The Aerospace Corporation, El Segundo, CA. 2013.

*MIL-STD-810G. Department of Defense Test Method Standard for Environmental Engineering Considerations and Laboratory Tests. 2008.

European Space Standards and Recommended Practices

Consultative Committee for Space Data Systems (CCSDS). http://public.ccsds.org/default.aspx. Select "publications" from the menu.

Mission Assurance Policies and Directives

DOD policy, directives, handbooks, and web resources can be found at: http:www.acq.osd.mil, select tab "Policy & Guidance."

CJCSI 3170.01. Joint Capabilities Integration and Development System. 2015.

DODI 5000.02. Operation of the Defense Acquisition System. 2015.

SMC policy, directives, instructions, and guides can be found on Aerospace Aerolink in the Mission Assurance Portal and on SMC Livelink in the Process Asset Library (PAL), including the following MA-related guidance:

SMCI 62-109. Configuration Management. 2015.

SMCI 62-104. Space Systems Software Acquisition and Process Improvement. 2009.

SMCI 63-1207. Programmatic Environment, Safety, & Occupational Health Evaluation (PESHE). 2013.

SMC-G-001. SMC Life Cycle Systems Engineering. 2009.

SMC-G-002. Reliability and Maintainability. 2009.

SMC-G-004. In-Plant Resources. 2010.

SMC-G-007. Mission Assurance Tailoring Guide. 2013.

SMC-G-009. SMC Hazardous Material Management Program Guide. 2013.

SMC-G-1201. Assurance of Operational Safety, Suitability, and Effectiveness (OSS&E). 2009.

SMC-G-1202. Space Flight Worthiness Criteria (SFWC). 2009.

SMC-G-1203. Independent Readiness Review Process. 2009.

SMC-G-1205. SMC Risk Management Process Guide. 2013.

Handbooks, Guides, and Books

Listed are guides and handbooks developed by The Aerospace Corporation, external government agencies (e.g. DOD, USAF, SMC, and NASA), and the professional engineering community for systems engineering that have applicability to satellite ground segment acquisition and operation.

Abelson, Linda A., et al., *Software Measurement Standard for Space Systems.* TOR-2009(8506)-6, The Aerospace Corporation, El Segundo, CA. May 5, 2011.

Abelson, Linda A., Richard J. Adams, Alan B. Arehart, Myron J. Hecht, Leslie J. Holloway, David J. Naiditch, and Robyn M. Wilkes, *Integrating Software Topics Into the Request for Proposal*, TOR-2011(8506)-117, The Aerospace Corporation, El Segundo, CA. July 19. 2012

Adams, Richard J. and Suellen Eslinger, *Software Sustainment Guidance*, TOR-2013-00693, The Aerospace Corporation, El Segundo, CA. December 31, 2013.

Body of Knowledge and Curriculum to Advance Systems Engineering Project (BKCASE). *Systems Engineering Body of Knowledge*. Version 1.5.1. 2015 [http://sebokwiki.org/]

Chiulli, Roy M. ISO 9000: An Aerospace Engineer's Handbook for Implementing International Standards for a Quality Systems. The Aerospace Press, El Segundo, CA. 2002.

Chrissis, Mary Beth, Mike Konrad, and Sandy Shrum. *CMMI® for Development; Guidelines for Process Integration and Product Improvement*, Version 1.3, Third Edition, Addison-Wesley 2011.

Defense Acquisition University. *Defense Acquisition Guidebook*. [https://dag.dau.mil]

Dixon, James M., Christine M. Rink, Craig V. Sather. *Digital ASIC/PLD Development Handbook for Space Systems*, TOR-2006(3904)-1. The Aerospace Corporation, El Segundo, CA. 2006.

Elbert, Bruce. The Satellite Communication: Ground Segment and Earth Station Handbook. 2nd edition. Boston: Artech House. 2014.

Englehart, William C. *Space Vehicle Systems Engineering Handbook*. TOR-2006(8506)-4494, The Aerospace Corporation, El Segundo, CA. 2005.

Eslinger, S., L. J. Holloway, and R. M. Wilkes, *Space Segment Software Readiness Assessment*, TOR-2011(8591)-20, The Aerospace Corporation, El Segundo, CA. June 3, 2011.

Gallagher, B. P., M. Phillips, K. Richter, and S. Shrum. *CMMI® for Acquisition; Guidelines for Improving the Acquisition of Products and Services*, Ver 1.3, 2nd Edition. Addison-Wesley. 2011.

Guarro, Sergio B., Gail A. Johnson-Roth, William F. Tosney. *Mission Assurance Guide*. TOR-2007(8546)-6019 Rev B, The Aerospace Corporation, El Segundo, CA. 2012.

Helvajian, Henry and Siegfried W. Janson, eds. *Small Satellites – Past, Present and Future*. The Aerospace Press, El Segundo, CA. 2008.

International Council on Systems Engineering (INCOSE). *Systems Engineering Handbook*. Version 4.0. Hoboken, NJ: John Wiley. 2015.

Martin, Donald, Paul R. Anderson, Lucy Bartamian. *Communication Satellites, Fifth Edition*. The Aerospace Press, El Segundo, CA. 2007.

Military Handbook: Design, Construction, and Testing Requirement for One of a Kind Space Equipment. DOD-HDBK-343 (USAF). February 1, 1986.

National Aeronautics and Space Administration (NASA*). NASA Systems Engineering Handbook*. SP-2007-6105 Rev 1. 2007.

National Research Council. Pre-Milestone A and Early-Phase Systems Engineering: A Retrospective Review and Benefits for Future Air Force Acquisition. Washington, DC. The National Academies Press, 2008.

Naval Systems Engineering Group Naxial. *Systems Engineering Guide*, October 2004.

Office of the Assistant Secretary of Defense for Systems Engineering. Department of Defense Risk, Issue, and Opportunity Management Guide for Defense Acquisition Programs. Washington, DC. 2015.

Office of the Staff Judge Advocate, Space and Missile Systems Center. Technical Data and Computer Software Rights, 7th Ed. *Acquiring and Enforcing the Government's Rights in Technical Data and Computer Software Under DoD Contracts*. 2015.

Owens, Karen L. and Joanne M. Tagami, *Recommended Software-Related Contractor Deliverables for National Security Space System Programs*. TOR-2006(8506)-5738, The Aerospace Corporation, El Segundo, CA. February 14, 2008.

Project Management Institute. A Guide to the Project Management Body of Knowledge (PMBOK Guide). Fifth Edition. 2013.

Project Management Institute. Government Extension to the PMBOK Guide Third Edition. 2006.

Project Management Institute. Software Extension to the PMBOK Guide Fifth Edition. 2014.

Rainey, Larry B., ed. Space Modeling and Simulation Roles and Applications Throughout the System Life Cycle. The Aerospace Press, El Segundo, CA. 2004.

Secretary for the Air Force for Acquisition (SAF/AQ). *I.S. Air Force Weapon System Software Management Guidebook*. 2008.

Software Program Manager's Network. *The Program Manager's Guide to Software Acquisition Best Practices*, Version 2.31. Arlington, VA. 1998.

Tosney, William F., Gail A. Johnson-Roth. *Mission Risk Planning and Acquisition Tailoring Guidelines for National Security Space Vehicles*. TOR-2011(8591)-5, The Aerospace Corporation, El Segundo, CA. 2010.

United States Air Force Space and Missile Systems Center. *Specialty Engineering Disciplines*. Vol 2, 1st Edition. 2011.

United States Air Force Space and Missile Systems Center. *Systems Engineering Primer and Handbook.* Vol 1, 4th Edition. 2013.

United States Air Force Space and Missile Systems Center. *Systems Engineering Critical Process Assessment Tools*. Vol 3, 1st Edition. 2012.

Wertz, James R., Wiley Larson. Microcosm, Inc, *Space Mission Analysis and Design*. Springer, Netherlands. 1999.

White, Julie D. and Lindsay G. Tilney. *The Test Like you Fly Process Guide for Space, Launch, and Ground Systems*.TOR-2014-02537, The Aerospace Corporation, El Segundo, CA. 2014.

White, Julia D., Geoffry A. Larson, Dan W. Hanifen. *Space Vehicle Test and Evaluation Handbook 2nd Edition*. TOR-2011(8546)-6018, The Aerospace Corporation, El Segundo, CA. 2012.

Mission Assurance Improvement Workshop — Best Practices

The U.S. space community (industry, academia, and government) formed the Mission Assurance Improvement Workshop (MAIW) to explore and document best practices and craft a common approach to mission assurance for the U.S. space program. The MAIW is a U.S space program community of practice dedicated to the development and promulgation of proven scientific, engineering, quality, and program management practices related to the U.S. space program's mission success. The following best practices documents are identified as a subset of the MAIW in terms of applicability to a ground segment.

Aguilar, Joseph A. *Design Assurance Guide*. TOR-2009(8591)-11, The Aerospace Corporation, El Segundo, CA. 2009.

Baxter, Michael J. *Guidance for Space Program Modeling and Simulation*. TOR-2010(8591)-17, The Aerospace Corporation, El Segundo, CA. 2010.

Betz, Frank C. *Space Segment Information Assurance Guidance for Mission Success*. TOR-2011(8591)-22, The Aerospace Corporation, El Segundo, CA. 2011.

Cantrell, John C. *Suggested Checklist to Improve Test Performance in the System Test Equipment Area*. TOR-2009(8591)-12, The Aerospace Corporation, El Segundo, CA. 2009.

Childers, Kenneth R. *Mission Assurance Practices for Satellite Operations*. TOR-2013-00293, The Aerospace Corporation, El Segundo, CA. 2013.

Covington, Richard K. *Design Review Improvement Recommendations*, TOR-2015-02545, The Aerospace Corporation, El Segundo, CA. 2015.

Duphily, Roland J. *Root Cause Investigation Best Practices Guide*. TOR-2014-02202, The Aerospace Corporation, El Segundo, CA. 2014.

Hecht, Thomas C. *Failure Review Board Guidance Document*. TOR-2011(8591)-19, The Aerospace Corporation, El Segundo, CA. 2011.

Hsu, Andrew Y. and Amy Weir. *Technical Risk Identification at Program Inception*. TOR-2014-02201. The Aerospace Corporation, El Segundo, CA. 2014.

Meshel, David C. *Counterfeit Parts Prevention Strategies Guide*. TOR-2014-02200, The Aerospace Corporation, El Segundo, CA. 2014.

Metodi, Tzvetan S. *Space Vehicle Testbeds and Simulators Taxonomy and Development Guide*. TOR-2010(8591)-16, The Aerospace Corporation, El Segundo, CA. 2010.

Speece, Dana J. and Rosalind Lewis. *Objective Reuse of Heritage Products*. TOR-2013(3090)-1, The Aerospace Corporation, El Segundo, CA. 2012.

Tosney, William F., Paul G. Cheng, Jeff B. Juranek. *Guidelines for Space Systems Critical Gated Events*. TOR-2009(8583)-8545, The Aerospace Corporation, El Segundo, CA. 2008.

White, Julia D. *Test Like You Fly Assessment and Implementation Process*. TOR-2010(8591)-6. The Aerospace Corporation, El Segundo, CA. 2010.

Acknowledgments

The Ground Segment Systems Engineering Guide was created by many authors and reviewed by many subject matter experts through The Aerospace Corporation. Geraldine Chaudhri and Gail Johnson-Roth co-coordinated the chapter definition and managed the content input by initially inviting authors to draft chapters, soliciting program office review as each chapter was submitted, and reviewed and edited material with technical publications to ensure content was accurate and complete. This handbook would not have been possible without the excellent technical editing provided by Karyn Zafran, and word processing by Mary T. Villanueva. Senior advisors, reviewers, and contributing editors including William Tosney, Rand Fisher, …., all of whom provided invaluable direction, advice, review, and feedback. The contribution authors and reviewers are acknowledge in Table I. Each chapter also acknowledges the specific authors at the beginning of their respective chapters.

Sergio J. Alvarado	Mission Futures, Ground Engineering
James V. Anderson	Space Superiority Systems Directorate, Space Program Operations
Alan B. Arehart	Systems Engineering and Development, Space Superiority Systems
David W. Bart	Independent Readiness Development, Engineering and Integration Division
James W. Boswell	Systems Engineering Imagery Programs Division
Leia R. Bowers	Software Acquisition and Modeling Department, Software Acquisition and Process Department
Laureen L. Branting	Mission Control, Ground Engineering
Steven K. Brownell	Program Engineering, Communication Programs
Daniel J. Byrne	System Integration and Test Office
Asya Campbell	Enterprise Ground Services, Developmental Planning and Architectures

Michael L. Campbell	Computer Applications and Assurance Subdivision, Computers and Software Division
Benjamin C. Cano	Space Assurance, Navigation Division
Erin Y. Carraher	Navigation and Geopositioning Systems Department, System Analysis and Simulation Subdivision
Tony B. Carwile	Systems Integration and Test Office, Mission Assurance Subdivision
J. Denise D. Castro-Bran	Systems and Operations Assurance Department, Mission Assurance Subdivision
Geraldine A. Chaudhri	Software Systems Assurance Department, Computer Applications and Assurance Subdivision
Jya-Syin W. Chien	Acquisition Risk and Reliability Engineering Department, Mission Assurance Subdivision
K. Rex Childers (casual)	Operations and Sustainment, MILSATCOM Division
Mel M. Cutler	Computer Applications and Assurance Subdivision, Computers and Software Division
Marc D. DiPrinzio	Mission Analysis and Operations Department, System Analysis and Simulation Subdivision
Marilyn K. Dubas (casual)	Software Engineering Subdivision, Computers and Software Division
Douglas Duncan	MILSATCOM Division (Civilian USAF)
Roland J. Duphily	Acquisition Risk and Reliability Engineering Department, Mission Assurance Subdivision

Robert B. Dybdal	Communication Systems Implementation Subdivision, Communications and Cyber Division
Suellen Eslinger (casual)	Software Engineering Subdivision, Computers and Software Division
Ricardo J. Espindola	Flight Operations Integration and Engineering, Space Innovation Directorate
Daniel P. Faigin	Validation and Requirements, Cyber Acquisition and Validation Department
Rand H. Fisher	Systems Planning, Engineering, and Quality, The Aerospace Corporation
Mary Jo Gura	Software Acquisition and Modeling, Software Engineering Subdivision
Thurman R. Haas	Systems Engineering Division, Engineering and Technology Group
Robyn L. Haleski	Software Acquisition and Modeling Department, Software Engineering Subdivision
Myron J. Hecht	Software Acquisition and Modeling Department, Software Engineering Subdivision
Sonia R. Henry	Acquisition Support and Information Department, Acquisition Analysis and Plan Subdivision
Leslie J. Holloway	Retired, The Aerospace Corporation
Dana L. Honeycutt	Technical Training and Development Department, The Aerospace Institute
Sharon K. Hoting (casual)	Engineering and Integration, Space Based Sensing Division
Daniel X. Houston	Software Acquisition and Modeling Department, Software Engineering Subdivision

Andrew Y. Hsu	Acquisition Risk and Reliability Engineering Department, Mission Assurance Subdivision
Martha I. Johnson	Software Acquisition and Modeling Department, Software Engineering Subdivision
Gail Johnson-Roth	Enterprise Systems Engineering, Corporate Chief Engineer's Office
William V. Kaida	Western Range, Launch and Satellite Control Division
Judy S. Kerner	Software Engineering Subdivision, Computer and Software Division
Brad Kizzort	Harris Corporation
Charles H. Lavine	Acquisition and Accreditation, Cyber Acquisitions and Validation Department
Yum Tong Lee	Acquisition Risk and Reliability Engineering Department, Mission Assurance Subdivision
Gretchen Lindsay	System Engineering Acquisition and Operations Department, Computers Applications and Assurance Subdivision
Gregory Lockwood	Software Systems Assessment, Software Systems Assurance Department
Sheryl M. Luera	Program Executability, Engineering and Integration Division
Robert E. Markin	Mission Analysis and Operations Department, System Analysis and Simulation Subdivision
David H. McCasland	Systems Integration and Test Office, Mission Assurance Subdivision

Arthur L. McClellan	Systems and Operations Assurance Department, Mission Assurance Subdivision
Rebecca McKenna	Mission Assurance and Development, Enterprise Engineering
Justin F. McNeill	JPL and Robotics Programs, Systems and Technology Programs Directorate
Suzanne P. Menichiello	Ground Systems Engineering, Systems Engineering Acquisition and Operations Department
Steven A. Meyers	Software Acquisition and Modeling Department, Software Engineering Subdivision
Lawrence H. Miller	Software Engineering Subdivision, Computers and Software Division
Rachel D. Morford	Future and International Programs, MILSATCOM Division
David J. Naiditch	Software Acquisition and Modeling Department, Software Engineering Subdivision
Rhoda G. Novak (casual)	Model Based Systems Engineering Office, Architecture and Design Subdivision
Randall M. Onishi	Advanced Demonstrations, MILSATCOM Division
Wayne T. Otsuki	Satellite Control Network Development, Range and Satellite Control Enterprise
Karen Owens	Retired, The Aerospace Corporation
Gary C. Palosaari	Software Acquisition and Modeling Department, Software Engineering Subdivision
Marybeth S. Panock	Validation and Requirements, Cyber Acquisition and Validation Department

Chandrakan C. Patel	Systems and Operations Assurance Department, Mission Assurance Subdivision
Dewanne M. Phillips	Enterprise Systems Engineering, Corporate Chief Engineering Office
John F. Reeves	Systems Software and Tools, Software Systems Analysis Department
Brian E. Shaw	Engineering Department, Engineering and Integration Division
James A. Shneer (casual)	Space Superiority Systems Directorate, Space Support Division
Dana J. Speece	Corporate Quality Management Office, Corporate Chief Engineer's Office
Timothy J. Spinney	Flight Operations Integration and Engineering, Space Innovation Directorate
Richard L. Staats	Mission Operations and Facilities, Infrastructure Services and Operations Directorate
Stephen A. Stoops	Infrastructure Services and Operation Directorate, Ground and Communication Division
Robert Sudakow (casual)	Systems Integration and Test Office, Mission Assurance Subdivision
Lindsay Tilney	Systems Integration and Test Office, Mission Assurance Subdivision
William F. Tosney	Corporate Chief Engineer's Office, System Planning, Engineering and Quality
Donald E. Town	Retired, The Aerospace Corporation
Bonnie R. Troup	Operations and Sustainment, MILSATCOM Division

Alan D. Unell	Computers and Software Division, Engineering and Technology Group
Mary T. Villanueva	Technical Publications, Internal Communications
Andrew E. Walther	Cyber and Systems Effectiveness, MILSATCOM Division
C. Jean J. Wang	Software Acquisition and Modeling Department, Software Engineering Subdivision
Marilee J. Wheaton	Systems Engineering Division, Engineering and Technology Group
Julia D. White	Systems Integration and Test Office, Mission Assurance Subdivision
Scott C. Wilkes	Software Systems Assessment, Software Systems Assurance Department
Wayne G. Yenne	Computer Applications and Assurance Subdivision, Computers and Software Division
Fitzroy E. Younge, Jr. (casual)	Independent Readiness, Engineering and Integration Division
Karyn C. Zafran	Technical Publications, Internal Communications

Contents

Figures

Tables

Chapter 1
Ground Segment Overview

Tony B. Carwile
Systems Integration and Test Office
Mission Assurance Subdivision

Herein we describe how the ground segment fits into the space mission architecture, and present a reference architecture that includes important internal and external interfaces. This chapter also presents the general ground segment concept of operations, and discusses the variety of missions that are managed and the types of mission data that are processed.

1.1 Introduction

The ground segment is an essential component of a space mission architecture. The ground segment consists of terrestrial facilities and equipment used to conduct space operations for a range of communications, earth monitoring, and other satellite missions orbiting at different altitudes and inclinations. The ground segment is distinct from the launch, space, and user segments, but typically has cryptographically-secure interfaces to each segment to ensure data link integrity.

Figure 1-1 shows a ground segment reference architecture that depicts key internal infrastructure and external interfaces.

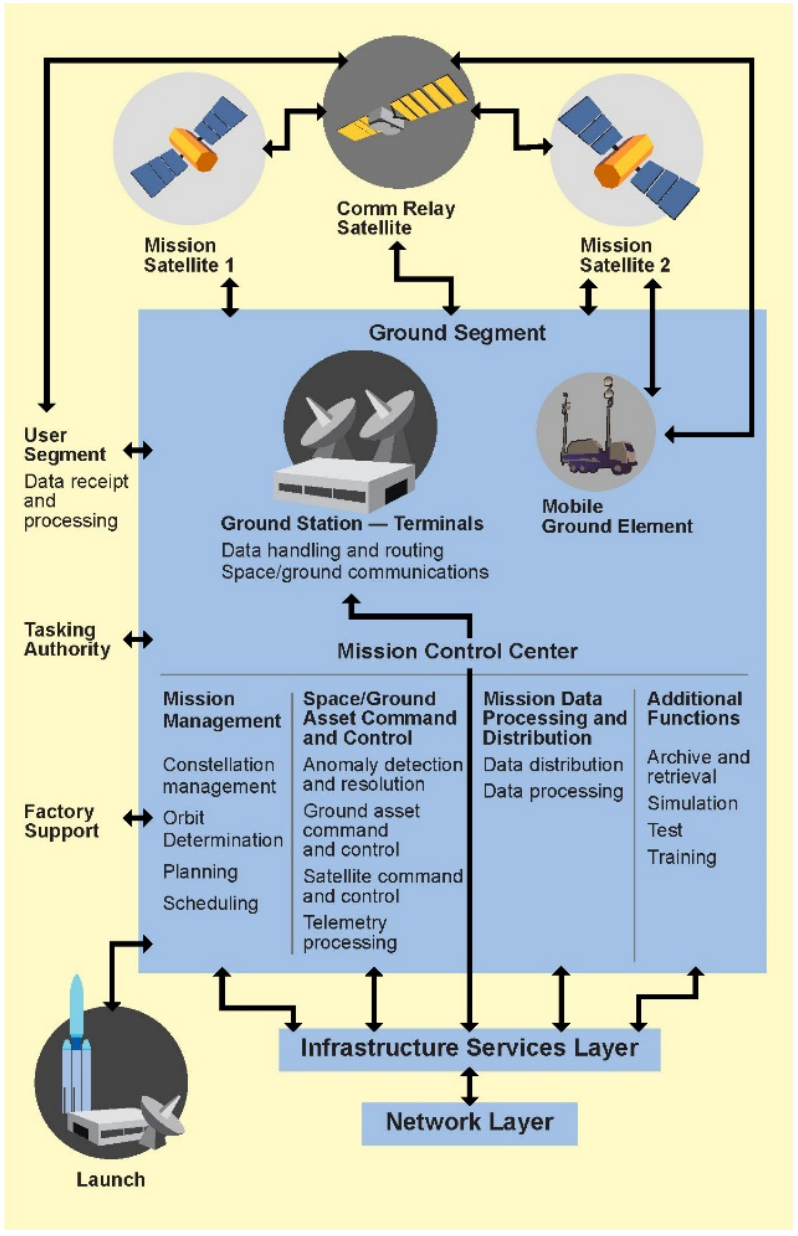


Figure 1-1. Representative ground segment reference architecture with external interfaces.

A ground segment generally is dedicated to support a single mission category type, although segments can be tailored to accommodate primary, secondary, and other hosted payload missions. The variety of satellite missions include communications (to provide immediate, world-wide broadcast, voice, or duplex data capabilities); Earth observation (weather, mapping); science (exploration, space weather observation, new technologies testing); geolocation services (global positioning); surveillance and reconnaissance; and space situational awareness.

Satellite categories can also have mission-unique payload data types, formats, and communication methods that require distinctive ground segment element characteristics; high-resolution imagery or signal collection missions may require extreme wideband/high data rate processing capabilities, and communications missions may implement exclusive anti-jamming features that require special signal handling methods and devices. Different satellite orbits can also affect ground segment element acquisition and design; a ground terminal developed to track a low-Earth orbiting (LEO) satellite may not be appropriate for tracking a geosynchronous Earth orbit (GEO) satellite (minute GEO antenna pointing changes over long durations can burn out LEO tracking antenna motors), and GEO tracking antennas may not be agile enough to track a LEO satellite.

Unique among ground segments is the Air Force Satellite Control Network (AFSCN), which provides common telemetry, tracking, and commanding (TT&C) services and shared resources to many different military satellite programs. Each mission using AFSCN resources must adhere to a well-defined set of interfaces and protocols, and follow established certification and approval processes to ensure program compatibility. Some programs maintain dedicated ground segments but may utilize the AFSCN for launch support and other services.

A major ground segment initiative is to increasingly automate routine functions and processes to allow operators to focus more on delivering timely mission products and effects. Common ground segment functions may also migrate to virtual and cloud-based environments, although characteristics and features of some mission-unique processes may limit comprehensive redeployments. Enterprise level TT&C solutions that provide a standard look and feel (similar to successful commercial sector implementations) currently are being investigated and assessed; this approach may yield annual operations and maintenance cost savings while increasing ground segment support capacity and capabilities.

1.2 Definitions

There are a number of common ground segment terms, acronyms, and related expressions that may be encountered in subsequent sections. A sample of these terms include:

Air Force Satellite Control Network (AFSCN) A global network of resources to provide TT&C support for launch and space vehicles.

Commanding The process of sending a formatted data structure to a bus or payload to perform a specific operation.

Communications security (COMSEC) The protection resulting from all measures designed to deny unauthorized persons valuable information, which experts in electronics or telecommunications might be able to find. COMSEC capabilities are used to protect information transiting terminal devices and transmission media from adversary exploitation to include transmission security capabilities designed to support operations security (OPSEC) and low probability of intercept/low probability of detection (LPI/LPD).

Concept of Operations (CONOPS) A verbal or graphic statement that clearly and concisely expresses what the joint force commander intends to accomplish and how it will be done using available resources.

Ephemeris A table or data file giving the calculated positions of a celestial object at regular intervals throughout a period.

Front end processor (FEP) A specialized hardware device with attendant software to format outgoing commands, and to synchronize/decommutate incoming telemetry frames and packets into convertible measurands/parameters for processing and display.

Ground segment (GS) The collection of physical, functional, and logical resources and capabilities used to manage and control a satellite mission.

Ground station-terminals The antennas, radio frequency (RF) and signal conditioning equipment, and supporting devices and infrastructure configured to track and communicate with a satellite.

Measurand Measurements taken from spacecraft on-board transducers or payload sensors for FEP conversion and display; also known as a parameter. A derived measurand is a special type that is created by combining multiple real satellite measurands together; combining single bit measurands into one derived measurand allows a more efficient monitor display.

Mission Control Center (MCC) The central location where satellite missions are managed and operated. MCC is used interchangeably with Satellite Operations Center (SOC).

Module (MOD) Term commonly used interchangeably with SOC for satellite control facilities allocated to a mission area at Schriever Air Force Base.

Operations security (OPSEC) A process of identifying critical information and subsequently analyzing friendly actions attendant to military operations and other activities.

Payload The components performing the satellite's mission (for example, communications, navigation, weather, warning). A satellite can carry more than one payload, expanding its primary mission or giving it secondary missions.

Payload operations center (POC) A central facility either remotely or co-located with a MCC where a payload can be independently managed and operated. A POC is most typically used to support hosted payloads. The POC may utilize a separate "out-of-band" RF communications link with the payload (the main satellite TT&C link is termed "in-band").

Ranging The ground station/terminal capability to determine the precise distance between a tracked satellite and the ground terminal at specific times and intervals. Ranging data is used to calculate a satellite ephemeris.

Remote ground facility (RGF) A facility remotely located from an operational control node that provides the satellite-to-ground TT&C interface. Functionally equivalent to a ground terminal (GT).

Remote tracking station (RTS) A shared AFSCN Range Segment GT resource configured under the real-time control of a user that provides telemetry, tracking, commanding, and mission data retrieval services for assigned satellites.

Satellite operations center (SOC) Facility conducting prescribed on-orbit TT&C activities for operational satellites under combatant command (COCOM) authority. Activities include, but are not limited to, recovering mission data, monitoring satellite status and safety, maintaining bus and payload capabilities, and maneuvering and stationkeeping the satellite throughout its mission lifetime.

Space operations (SOPS) The mission area encompassing space control, space surveillance, missile warning, satellite operations and spacelift.

Space vehicle (SV) Although usually considered interchangeable with satellite, in some program instances SV is used to indicate only the bus without the associated payload, while satellite is defined as the bus plus all payloads.

Telemetry Commutated measurements taken from spacecraft on-board transducers or payload sensors and transmitted to the ground station in defined frame or packet formats for conditioning, relay, and processing. Two standard satellite telemetry types are payload (e.g. mission) and satellite state-of-health (SOH) data.

Telemetry, tracking, and commanding (TT&C) The common ground segment functional term that encompasses receiving satellite telemetry data, performing antenna tracking of the satellite, and transmitting commands to the satellite.

Tracking The ground terminal function to acquire (i.e. locate) a satellite and continuously follow the satellite movement to allow the ground segment to send commands and receive telemetry. The ground terminal tracks the main satellite downlink beam either automatically or by a preprogrammed sequence.

Transmission security (TRANSEC) The component of COMSEC that results from all measures designed to protect transmissions from interception and exploitation by means other than cryptoanalysis.

1.3 Ground Segment Description

The ground segment reference architecture shown in Figure 1-1 is a general representation of the elements, functions, and interfaces required to manage and operate a satellite mission. The ground segment includes the facilities and infrastructure, antennas, communications interfaces, and the computer hardware and software used to plan and execute the range of satellite operations throughout all mission phases.

Primary physical elements of the ground segment are the MCC, where satellite commanding and telemetry processors, monitors, and associated off-line and real-time applications are installed and operated; the ground station-terminals, where the antennas and supporting equipment used to transmit, receive, condition, and route data are located; and the mobile ground element, which replicates a limited set of the MCC and ground station-terminals capabilities.

The MCC is the primary location where daily satellite operations are conducted. Normal day-in-the-life (DITL) operations usually involve performing a standard sequence of regular SOH and mission-unique tasking, planning, scheduling, and real-time execution routines; abnormal/non-nominal operations can entail

executing “safing” procedures to protect a satellite’s health and safety due to a detected anomalous condition.

For a typical mission, the MCC must receive prioritized requests from the tasking authority (or approved channel assignments and schedules for a communications mission), calculate satellite ephemerides, generate a satellite command plan (based on acceptable spacecraft visibilities, power constraints, and other requirements and limitations), and request/schedule/assign ground segment resources. MCC operators then conduct real-time satellite contact operations at assigned times by configuring allocated ground segment resources, sending the commands listed in an approved command plan to the satellite through the assigned ground terminal, monitoring processed packet- or frame-based telemetry to confirm proper command receipt/execution and satellite SOH, and acquiring and distributing data to authorized recipients. Minor variations in this general execution flow may occur based on satellite orbit, mission type, mission sponsor, and on-board capabilities.

Air Force program MCCs are primarily located at Schriever Air Force Base and utilize shared AFSCN RTS resources, while other programs have separate dedicated facilities. Satellite manufacturers may also maintain active centers with some off-line and real-time capabilities to support certain vehicle engineering support events.

The ground station-terminals element is where space to ground communications take place; the element may be co-located with the MCC or at a remote site. The ground station-terminals element consists of antennas and supporting equipment used to transmit, receive, condition, and route data to other parts of the ground segment. Key ground station-terminals functions involve TRANSEC encryption/decryption, data handling (e.g. packetizing/depacketizing) and routing, and signal up/down frequency conversions, amplification, and conditioning/formatting of both transmitted satellite commands and received telemetry.

Mobile Ground Elements (MGEs) are ground segment assets that can be deployed to an operational theater. They can have several distinct configurations based on program needs. An MGE can operate as a backup MCC and ground station-terminals combination, replicating key functional capabilities of the primary MCC in case of catastrophic failures. MGEs are typically built as transportable vans capable of being airlifted to a designated installation site.

Ground segment internal activities are functionally organized into off-line (i.e. non-realtime) and real-time categories. Off-line categories shown in Figure 1-1 are the Mission Management, Additional Functions, and Infrastructure Services activities; real-time categories are the Space/Ground Asset Command and Control, Mission Data Processing and Distribution, Data Handling and Routing,

and Space/Ground Communications activities. The underlying Network Layer shown in Figure 1-1 is the logical element that allows the listed individual activities to pass or share data.

MCC Mission Management functions include:

- Receiving mission tasking
- Planning satellite orbit and attitude adjustments
- Generating satellite ephemerides for orbit determination and/or ephemeris propagation
- Planning payload tasking
- Deconflicting tasks and ground resources
- Scheduling ground resources
- Approving mission plans

MCC Additional Functions include:

- Archiving and retrieval of satellite telemetry and tracking data
- Training operators
- Simulation and test

MCC Space/Ground Asset Command and Control functions include:

- Configuring, controlling, and monitoring ground equipment
- Formatting and transmitting satellite commands
- Verifying satellite command execution
- Decommutating, processing, displaying, analyzing, and trending telemetry
- Responding to reported alarms/events
- Detecting satellite anomalies and engaging in analysis/resolution activities
- Developing, testing, approving, and executing operational procedures

MCC Mission Data Processing and Distribution functions include:

- Configuring mission data processing equipment
- Receiving, processing, distributing, and archiving mission data
- Generating and distributing mission data reports

Infrastructure Services functions include:

- Managing databases
- Managing networks and interfaces
- Maintaining hardware and software
- Monitoring and maintaining information assurance security

The Network Layer functions include:

- Providing interprocess communications capabilities between ground segment elements

A typical mission CONOPS involves sequentially performing the listed off-line, pre-contact Mission Management functions, followed by the Space/Ground Asset Command and Control and Mission Data Processing and Distribution real-time contact functions. Off-line Additional Functions and the Infrastructure Services functions are predominantly administrative and logistical activities that support these mission tasks.

Figure 1-1 is a representative ground segment reference architecture that provides a context for the information presented in later document sections. Because individual program missions, satellites, payloads, and orbits are quite varied, a program's ground segment architecture must be designed and tailored to accommodate specific mission requirements.

1.4 Technical Considerations

The variety of space missions creates special needs for a corresponding variety of ground segment capabilities. Some critical missions have extremely short reporting timeliness requirements to provide real-time situational awareness to cognizant users; this need demands that the ground segment have low-latency communications paths and ground station high-rate processors. Other missions have significant downlink data rates, which require substantial wideband processing capabilities at the ground station-terminals. And yet other missions demand coordinated satellite timing synchronization, which necessitates an exacting and accurate timing source and known deterministic data path latencies.

Many legacy ground segment architectures are considered "stovepipe" designs because of their lack of modularity and expandability, tight hardware/software integration, and single mission support. This design inflexibility creates serious technical and programmatic challenges and can contribute to significant cost impacts and logistical efforts to maintain system currency without affecting ongoing operations.

Creating a modular ground segment architecture that can be adapted and shared to support a variety of uncommon mission requirements and types is a current design goal under consideration to reduce acquisition and sustainment costs. Service-oriented architectures (SOAs), cloud computing, and related business process workflow methods are all under detailed assessment to determine their applicability, efficiency, and effectiveness in addressing projected technical needs while satisfying future fiscal realities. Yet a SOA design approach creates its own set of challenges with the increasing number of information assurance/ cybersecurity threats and the utilization of a significant number of commercial off-the-shelf (COTS) products, which would require substantial coordinated integration testing to mitigate operational risks across multiple programs.

Future architectures must maintain an aggressive security posture to eliminate or mitigate potential ground segment vulnerabilities. Ground segment architectures must also embrace such design directions to accommodate future increases in satellite capabilities and future decreases in acquisition, operations, and support funding.

1.5 Programmatic Considerations

The acquisition process is defined as a sequenced series of time phases: the Material Solution Analysis phase, Technology Maturation and Risk Reduction phase, Engineering and Manufacturing Development phase, Production and Deployment phase, and Operations and Support phase. Experience has shown that many ground segment acquisitions have been challenged to meet their planned operational acceptance schedule due to unaddressed issues occurring in the different acquisition phases. Ground segment acquisition personnel must be alert to such issues, and address the root cause early to ensure a timely delivery.

On one program ground segment products and applications were acquired to replace a legacy system, but the acquisition contract did not include a schedule or a budget to actually implement the transition of the new hardware and software to operations. This issue could have been recognized and addressed in the Technology Maturation and Risk Reduction phase. The lesson learned is to contractually accommodate transition activities, provide sufficient funding, and clearly define contractor and government development, integration, test, and transition roles.

On more than one program ground segment and user segment incompatibilities were discovered late in the Production and Deployment phase. Contributing factors included insufficient intersegment coordination and communication due to asynchronous user and development schedules. For example, on one program user segment elements were delivered 3 years prior to the ground segment development, and subject matter experts had moved to other programs in the interval and were not available to clarify interface interpretations. In addition, a

lack of user community engagement during ground segment development led to user unpreparedness and resultant schedule delays. Root causes included inadequate end-to-end systems engineering, interface control document misinterpretations, and ineffective stakeholder participation during ground segment development.

Another common ground segment acquisition deficiency identified on several Air Force programs involved inadequate CONOPS/capabilities description documents prior to requirements development, which contributed to the generation of poor/untestable/volatile requirements, inadequate segment architectures, and insufficient test plans and procedures. This deficiency could have been recognized and corrected during the Technology Maturation and Risk Reduction Phase, but unrealistic development schedules and budgets coupled with a lack of effective and coordinated contractor oversight led to significant program delays.

Ground segment acquisitions can be developed, installed, and accepted on schedule and on budget; however, it requires rigorous, realistic program and test planning and design up-front, coherent and complete program documentation, an involved stakeholder community, and a proactive government program office that provides thorough technical and programmatic oversight to confirm that required mission capabilities will be delivered.

1.6 Bibliography

Air Force Space Command Instruction 10-1204 *Satellite Operations*, May 15, 2014 (AFSPCI10-1204).

AFSPC Guidance Memorandum (GM) 2015-13-01, Space Operations Crew Force Management, Training, Standardization and Evaluation, July 14, 2015.

Standardized Interface Specification (SIS) Between the Air Force Satellite Control Network (AFSCN) Network Management System (NMS) and Users, SIS-000509E Honeywell Technology Solutions, Inc. January 26, 2006.

Department of Defense Instruction (Interim) 5000.02, *Operation of the Defense Acquisition System*, January, 2015

Joint Publication 1-02, *Department of Defense Dictionary of Military and Associated Terms*, 8 November 2010 (As Amended Through November 15, 2015).

Joint Chiefs of Staff, United States Department of Defense, Joint Publication 6-0, *Joint Communications System*, June 10, 2015.

1.7 Acronyms

AFSCN	Air Force Satellite Control Network
COCOM	combatant command
COMSEC	communication security
CONOPS	concepts of operations
COTS	commercial off-the-shelf
DITL	day-in-the-life
FEP	front end processor
GEO	geosynchronous Earth orbit
GS	ground segment
GT	ground terminal
LEO	low Earth orbit
LPD	low probability of detection
LPI	low probability of intercept
MCC	mission control center
MGE	mobile ground elements
MOD	module
OPSEC	operations security
POC	payload operations center
RF	radio frequency
RGF	remote ground facility
RTS	remote tracking station
SOA	service-oriented architecture
SOC	satellite operations center
SOH	state of health
SOPS	satellite operations
SV	space vehicle
TRANSEC	transmission security
TT&C	telemetry, tracking, and commanding

Chapter 2
Ground Station Terminals

Robert B. Dybdal
Communication Systems Implementation Subdivision
Communications and Cyber Division

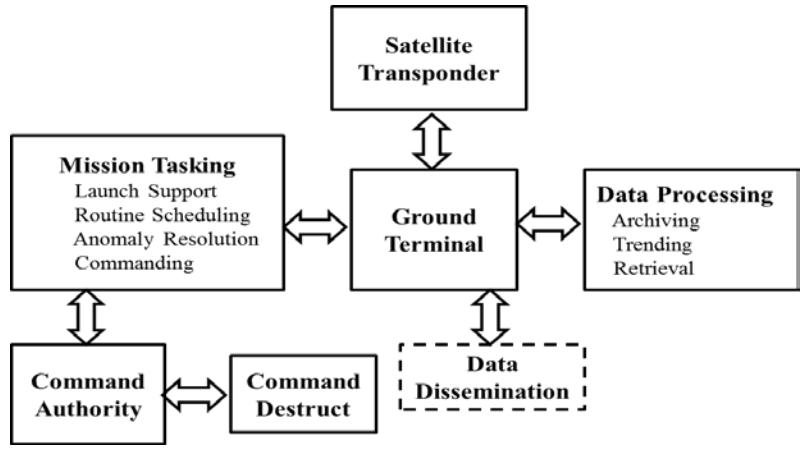
Ground station terminals are communication assets that provide essential support for satellite system operation, commanding, monitoring, and diagnostics. These are two types of ground terminal designs. The first ground terminal design provides satellite control for operational satellites commonly referred to as telemetry, tracking, and control (TT&C) terminals. The second ground terminal design provides mission support for specific program applications and provides more detailed and dedicated evaluation of payload operational performance and anomaly resolution capabilities. This chapter describes the system and subsystem requirements, development process, technologies, test verification methodologies, and supporting software interfaces.

2.1 Introduction

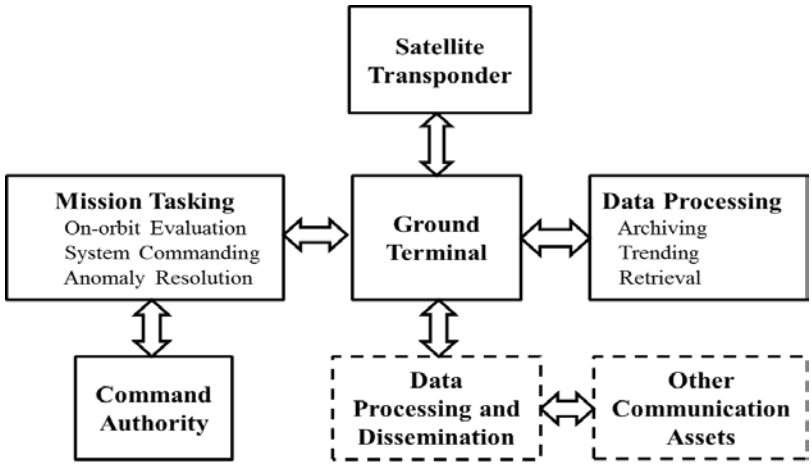
Satellite control terminals service a large number of satellites, and consequently their monitoring capabilities are necessarily more routine and focus on the overall satellite. Satellite health and status are principal concerns: operating temperatures, prime power voltages, battery charging status, attitude alignment, and other status parameters are routinely monitored. Satellite angle tracking and ranging capabilities provide inputs to Kalman filtering techniques, which together with other data, such as radar tracking data, are used to determine the satellites ephemeris, defining their orbital locations. Launch support is another major task that includes monitoring the assent and commanding initial on-orbit deployments. Satellite support systems communicate with relatively small data rates, but use large antennas to have ample margin to accommodate potentially degraded satellite transponders. Determining adequate margin requirements fosters considerable debate. These systems communicate using both space ground link subsystem (SGLS) [1] and unified S-band [2, 3, 4] modulation formats. Because their frequency allocations are shared with other applications, concerns of mutual interference arise. These concerns can be mitigated by limiting transmit power to levels consistent with link closure requirements and reducing antenna sidelobe levels away from the main beam. While power control can be implemented, existing physically large antennas enclosed by radomes limit the application of sidelobe control techniques that can be effective with smaller antennas.

Mission control terminals are dedicated to specific programs and provide a more detailed and dedicated evaluation of payload performance than satellite support systems and more extensive anomaly resolution capabilities than satellite support terminals. Mission control of navigation satellites, for example, upload clock time offsets and ephemeris updates to maintain user navigational accuracy. Remote sensing satellites download stored data to mission control stations for subsequent routing to processing facilities that produce and disseminate data products. Mission control stations for communication satellites evaluate and trend payload performance, allocate resources to system users, provide commanding that repositions antennas to satisfy changing coverage needs, and vary transponder gain values to respond to traffic variations.

The interfaces of these terminals with the overall system in Figure 2-1 are also important in understanding their roles and requirements. Both terminal designs communicate with the satellite's transponder and are directed by a command authority. Today's satellite transponders, like the support terminals, heavily rely on digital technology [5, 6]. The nature and amount of software continues to increase in satellite payloads; further attention is required to develop capabilities so that appropriate data rates are available for on-orbit software upgrades. Commercial communication satellites commonly use communication capabilities to load software upgrades. Mission tasking differs in emphasis between the two terminal designs following their roles and responsibilities. Satellite support systems provide overall satellite health and status data while mission support systems provide more detailed payload performance assessments and commanding. Telemetry data processing, trending, and archiving are important functions of both satellite control and mission control terminals. Initial on-orbit testing evaluates satellite system compliance, assesses both primary and redundant paths, demonstrates required commanding capabilities, and processes baseline measurements that are archived for comparison with subsequent data throughout the satellite's lifetime. Data trending during the satellite's life evaluates changes in system performance and capabilities. An important program responsibility is to devise and document flags that identify deviations from anticipated performance. Such flags prompt more detailed performance assessments and identify alternative capabilities to maintain system operation. Both terminal designs can be tasked to resolve on-orbit anomalies. In both cases, terminal built-in test equipment (BITE) capabilities have high importance so that identified anomalies can be distinguished between satellite malfunctions and terminal shortfalls.



(a) Satellite control terminal



(b) Mission control terminal

Figure 2-1. Control terminal interfaces.

While both terminal designs have overlapping and complementary objectives, their roles differ. The satellite control terminals also require a small communication system that provides a command destruct capability in the unhappy event of uncontrollable launch anomalies that require aborting the launch. Both terminal designs can process and disseminate data. The Defense Meteorological Satellite Program (DMSP) satellites, for example, use space-ground link system (SGLS) downlinks to directly downlink mission data to individual users and forward satellite stored data to data processing facilities

where data products are produced and disseminated. The National Oceanic and Atmospheric Administration (NOAA) satellites route mission data on separate downlinks to individual user terminals and to command and data acquisition (CDA) terminals for processing and dissemination for data product information. Communication satellite architectures can include gateway terminals where the uplink signal collection is processed and routed to downlink destinations and interfaces are provided to and from other communication assets.

2.2 Development Process

The overall ground terminal hardware development process for satellite support communication assets follows the process documented in Figure 2-2 [7]. The requirements stipulated in a technical requirements document (TRD) are used to develop a verification matrix that identifies compliance verification approaches, namely inspection, analysis, demonstration, and test, which are identified to demonstrate TRD requirement satisfaction. Design tradeoffs are examined to select a detailed candidate configuration that is evaluated to assess system compliance. Detailed evaluation of subsystem requirements flowed down to determine required capabilities. Many subsystem components exist in commercial off-the-shelf (COTS) products and alternatives from available vendors are evaluated. Some components specific to the application will require development, e.g., filters, and requirements are written for their development to allow vendor selection. An important capability is adequate BITE that is essential to identify operational shortfalls and provide a prompt return to service to satisfy system availability requirements, particularly with today's trends for terminal remote operation. Test plans and procedures are then derived based on the verification matrix. The verification extends over demonstration, qualification, integration, and acceptance phases. A fourth phase, sustainment, results since the lifetime of these ground terminals often exceed decades and upgrades are required to satisfy evolving requirements and to replace equipment with maintenance and obsolescence issues.

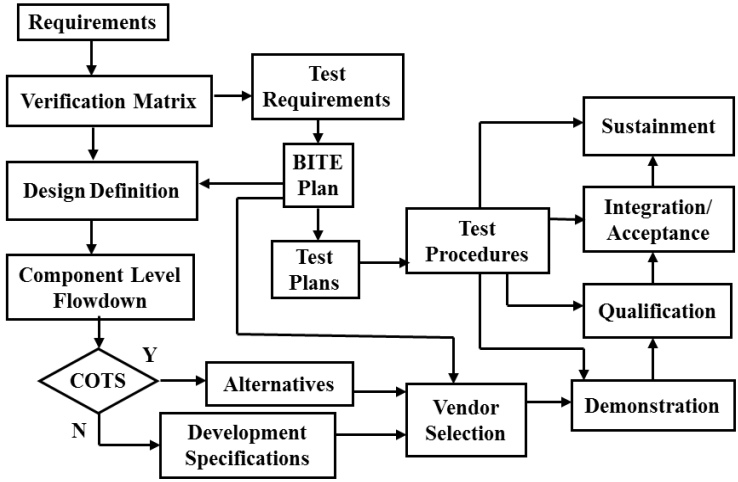


Figure 2-2. Ground terminal development process.

The process is accompanied by a series of reviews. A system requirements review (SRR) follows the completion of the verification matrix to ensure the requirements are fully understood and that the means to evaluate compliance are established. A preliminary design review (PDR) is held at the completion of the design definition to evaluate the design approach and approve plans for the COTS and development items, assess the vendor selection, and evaluate the perceived development risk. The requirement verification matrix is carefully examined during the PDR to determine system compliance and to assure test methodologies exist to satisfy the required verification. A critical design review (CDR) is held after the necessary development is completed to evaluate risk of compliance and to obtain approval to proceed with the integration and acceptance test phases. The requirement verification matrix is again examined to assure the design elements comply with TRD requirements and determine if the necessary test assets and procedures are in place to demonstrate the integrated system complies. The sustainment phase has a more limited evaluation pertaining to the replacement and evolved equipment and addresses the interface and component test requirements to assure reliable, compliant operational performance. Other reviews, manufacturing readiness reviews (MRR) and test readiness reviews (TRR) are conducted throughout the development to assure both component and integrated system elements can be implemented and tested.

2.3 System Component Requirements and Test Methodologies

The terminal hardware is comprised of the subsystems in Figure 2-3. Narrow beamwidth, high-gain antennas are required to satisfy performance objectives. The capability to dynamically track orbiting satellites typically uses closed-loop,

monopulse design processed by a tracking receiver. The antenna control unit commands the antenna's angular position initially in an open-loop manner based on the satellite ephemeris and terminal location having sufficient accuracy to allow closed-loop tracking systems to acquire and track satellites. The antenna is connected to the terminal's receiver and transmitter by a diplexer that provides sufficient isolation between the transmitter and receiver to avoid degrading communications. The transmitter modulates the information provided by the data source and upconverts to the transmit frequency, and amplifies its power to satisfy link closure requirements. Similarly, the receiver establishes the system noise temperature, demodulates the received signal, processes and then disseminates the information. The requirements, technology, and testing methodologies for each element are described in some detail.

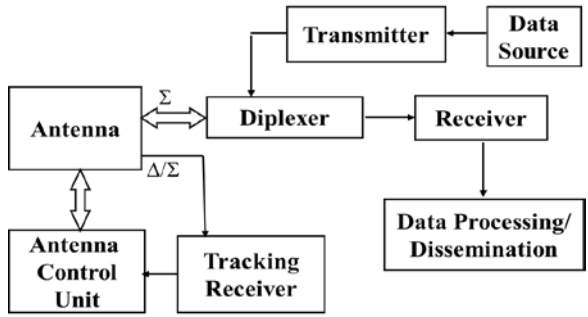


Figure 2-3. Basic ground terminal functional diagram.

2.4 Antenna

2.4.1 Requirements, Technology, and Testing

The antenna requirements span its radio frequency (RF) performance, pointing and tracking capabilities, power-handling limitations, and environmental issues. Reflector antenna technology is commonly used for both cost and performance reasons. The RF performance parameters include gain and pattern characteristics, the input impedance and insertion loss, polarization over the required receive and transmit frequencies, and transmitter power handling limitations. At a system level, two RF parameters define the required system link performance [7] whose values are derived from link analyses, as stipulated in the TRD.

The first system parameter is gain over temperature (G/T) that characterizes the system receive sensitivity that equals the antenna gain divided by the system noise temperature where both parameters must be referenced to the same system reference plane. The units of this parameter is dBi/K, the antenna gain dBi (dB relative to ideal isotropic coverage) divided by the system noise temperature

expressed in Kelvin. The system noise temperature is the sum of the antenna noise temperature and the receiver noise temperature discussed later. A convenient terminal plane is the receiver's low noise amplifier (LNA) input terminal that is typically measured to evaluate the receiver's compliance. The antenna gain at this reference plane must account for system losses in the antenna, its tracking circuitry, and diplexer. The antenna noise temperature results from ground emission and atmospheric loss and contributions from ohmic loss in the antenna and diplexer. Atmospheric loss increases at low elevation angles because of the increased atmospheric path length and ground emission contributions as the main beam approaches the horizon. Antenna noise temperature is referenced to a specified elevation angle, such as 20^{o}, to minimize emission contributions from local terrain obscura variations. An example analysis and measurement of antenna noise temperature is found in [8].

The second system parameter, effective radiated power (ERP), characterizes the system transmit performance. The ERP equals the transmit antenna gain multiplied by the transmitter power reduced by the loss between the transmitter and antenna. The units of this parameter are dBW, dB relative to 1 watt. The system loss includes the diplexer loss and losses in the connection between the antenna and the transmitter. The diplexer is connected to the antenna's feed output. The receiver's LNA and sometimes downconverter are directed connected to the diplexer and the output is routed to the terminal. The transmitter typically located in the positioner is connected to the diplexer with waveguide incorporating rotary joints to allow positioner motion and flexible waveguide sections to compensate coefficient of thermal expansion (CTE) differences.

The required antenna gain is derived from the system's G/T and ERP requirements. Generally, the antenna size is determined by the gain needed to satisfy G/T requirements and the ERP for that size antenna determines the required transmitter power. If excessive transmitter power levels result, the antenna size is increased to provide sufficient gain to obtain reasonable transmitter power levels and the G/T benefits from additional margin. When new system designs are being developed, first order estimates of antenna sizes for a given gain and the corresponding beamwidth are required. The peak antenna gain equals

$$G = \eta(\pi D/\lambda)^2 \qquad (1)$$

where η is the antenna efficiency (55% commonly assumed), D is the antenna diameter, and λ is the wavelength at the operating frequency f (equals c/f, where c is the speed of light). This gain level is at the antenna output and does not include filter and connection losses. The half power beamwidth, θ_{HP}, is the angular pattern width 3 dB lower than the peak gain and equals

$$\theta_{HP} = K\lambda/D \quad (2)$$

where K is commonly assumed to be 70 degrees. As the system design evolves, these first-order estimates are refined using available computer analysis codes and the gain levels are burdened by system losses in filters, waveguides, and cables that are later validated by measurement.

Satellite-support antennas are physically large and are commonly installed at fixed locations at operational sites. The antennas are typically enclosed by radomes that serve two purposes. The first purpose uses the radome to isolate the surface area of large antennas from wind loading that produces antenna pointing misalignments. The antenna and radome foundations must have sufficient mechanical isolation to prevent their coupling that would produce alignment perturbations. The second purpose protects terminal equipment from the external environment.

In some cases, smaller transportable antennas are used that are impractical to enclose with a radome. There are two types of transportable designs. One type transports the antenna to the site and secures it to a prepared concrete pad that is provided along with prime power and other facility accesses. The second transportable design mounts the antenna and system hardware on a trailer that can be deployed to locations on an as-needed basis. Such antennas require a trailer with sufficient stiffness, well-positioned outriggers, and sufficient positioner power so that antenna tracking is maintained under wind-loading conditions. The two alternatives are evaluated based on schedule urgency that favors trailer mounted designs and/or cost differences between the prepared concrete pad and an appropriate trailer design.

Satellite control terminals operate at L and S-band frequencies using SGLS and Unified S-band modulation, while mission control terminals require additional frequencies appropriate to frequencies used by their program's payloads. Earlier, the issue of the shared frequency allocation for satellite support systems was raised. Interference to and from other systems can be mitigated by reducing the antenna's wide angle sidelobes [7] using well-known design techniques. Sidelobe control techniques have limited application to existing terminals because of their large size, panelized reflector construction, prime focus designs, and scattering from space frame radome structural members. Such sidelobe control techniques can be more effectively applied to more modest size antennas capable of providing adequate performance for nominal TT&C operation. The design techniques include solid reflector surfaces to avoid backlobe increases from gaps between reflector panels, offset reflector designs to eliminate feed blockage contributions to sidelobes, and tunnels and shrouds to avoid feed radiation components to sidelobes.

Large antennas present measurement challenges. The common practice assembles and tests each antenna at the vendor location. Generally, the design development proceeds by using commonly available computer analysis codes that provide high-fidelity projections of antenna performance. The vendor capability includes the test measurement facilities and instrumentation that demonstrate the assembled antenna fully complies with the specified performance. Any discrepancies between the measurements of the particular antenna and those of previous builds, if available, together with the analysis are resolved to assure imperfections are not present in that particular antenna. Once the antenna's specified compliance has been established, the antenna is relocated at its operational site.

Vendor test facilities are not available at the terminal's site. Conventional far-field antenna testing requires a distance equal to $2D^2/\lambda$ separating the test source and the antenna under test to provide uniform test fields over the antenna, where D is the antenna diameter and λ is the test frequency's wavelength. Large antennas require an excessive separation distance that is limited by both line-of-sight constraints and ground multipath. The commonly used alternative uses the known flux density from such radio sources as Cassiopeia A [7, 9, 10]. The radio source technique provides a means to measure G/T that equals

$$G/T = 8\pi k(Y - 1)/(S/\lambda^2) \quad (3)$$

where k is Boltzmann's constant (198.6 dBm/Hz/K), Y is a Y factor that equals the noise power when the antenna pointed at the radio source divided by the noise power when the antenna pointed away from the radio source at the same elevation angle as the source, S is the flux density of the source, and λ is the wavelength at the operating frequency. Y factors are the ratio of noise powers where the higher power (hot) is divided by the lower power (cold), i. e., hot/cold, and apply to several applications.

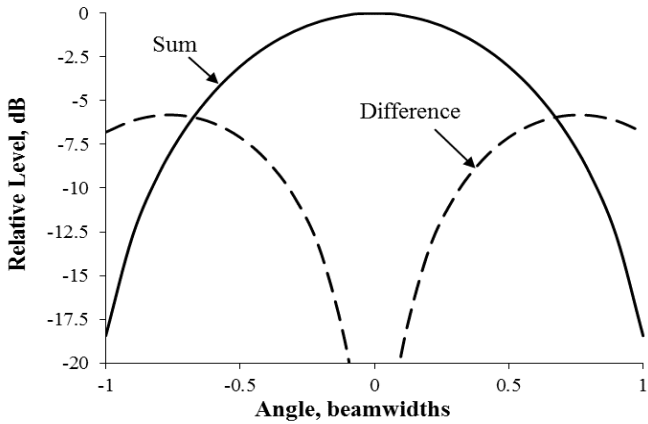
The antenna gain can be obtained from G/T values by separately measuring the system noise temperature T. The system noise temperature is the sum of the receiver and antenna noise temperatures at a common reference terminal. The receiver noise temperature T_{Rec} is obtained from a hot/cold Y factor measurement or noise figure meter measurements. The antenna noise temperature T_{Ant} is derived from another Y factor Y_{Ant} that equals the noise power when the receiver is terminated by a 290 K ambient temperature matched load divided by the noise power when the antenna is pointed at a particular elevation angle. Thus, Y_{Ant} equals $(290 + T_{Rec})/(T_{Ant} + T_{Rec})$, and the antenna noise temperature equals

$$T_{Ant} = [290 + T_{Rec}(1 - Y_{Ant})]/Y_{Ant} \quad (4)$$

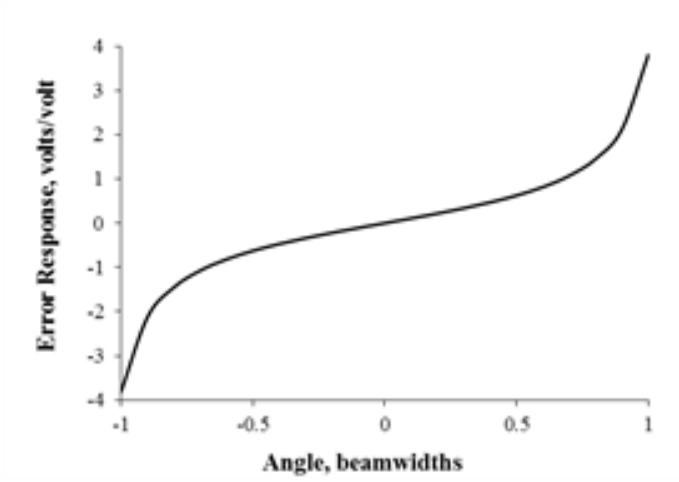
Once the antenna's noise temperature at a given elevation angle is determined, the antenna noise temperature's dependence on elevation angle is measured by repointing the antenna at different elevation angles and measuring the noise power. The antenna noise temperature's elevation variation equals the measured noise power variations divided by the noise power at the elevation angle used to measure the antenna noise temperature. The noise power at the elevation angle specified by the G/T requirement needs to be included in the data to determine G/T compliance. The transmit antenna gain is measured by replacing the transmitter with a suitable LNA, and measuring the G/T and the antenna and receiver noise temperature values. G/T measurements at the terminal site are compared to G/T measurements at the vendor's location to insure the antenna has been correctly reassembled at the site.

2.4.2 Antenna Pointing and Tracking

In operation, the antenna's main beam [7] must remain aligned with the satellite's direction. The antenna is initially open-loop pointed based on the satellite's ephemeris and the terminal's location on the Earth that determine the azimuth and elevation angles for the antenna's pointing. The receiver acquires the downlink signal transmitted by the satellite's telemetry transponder and the antenna pointing is refined by a closed-loop technique known as monopulse. Two types of antenna beams are formed. One is the sum beam, Σ, which is the normal pattern used for transmission and reception. The second type of pattern is a difference pattern, Δ, that by design has a pattern null coincident with the sum beam's axis, and is positive on one side and negative on the other side of the main beam axis. The ratio of the difference beam and the sum beam, Σ/Δ illustrated in Figure 2-1 is the error response. Example sum and difference patterns and the corresponding error response are illustrated in Figure 2-4. The error is zero when the antenna is properly aligned, positive on one side, negative on the other, and its magnitude measures how far the signal is removed from the main beam axis. In operation, these measurements are performed in orthogonal planes, e.g., azimuth and elevation, to align the antenna with the satellite's direction.



(a) Sum and difference patterns



(b) Error response

Figure 2-4. Monopulse patterns and error response.

The overall tracking accuracy requirements are based on the tolerable amount of signal loss caused by the antenna's misalignment with the signal. The peak levels of the antenna's main beam closely follows a Gaussian functional variation. The resulting pointing loss in Figure 2-5 illustrates the commonly used requirement that limits the pointing misalignment to 1/10 of a beamwidth to obtain pointing loss values less than roughly 0.1 dB.

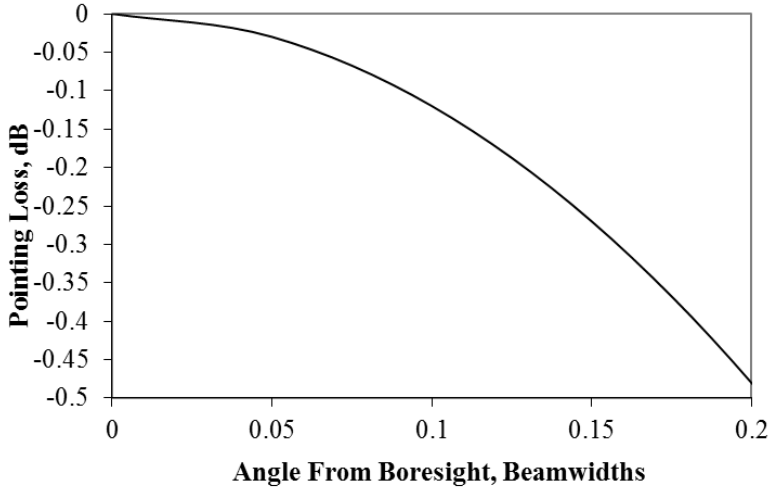


Figure 2-5. Pointing loss versus misalignment.

The testing addresses several elements of the antenna tracking capability [11]. The RF measurements include the peak gain levels, sum and difference patterns, and the error slope at both the feed and integrated antenna levels. The difference pattern is coupled onto the sum pattern, and the sum pattern is amplitude modulated by the difference pattern sampling. Ideally, the group delay between the sum and difference paths is identical. If this is not the case, the phase of the coupled signal varies over the bandwidth which degrades tracking performance. For narrow bandwidth applications, a phase shifter can compensate the group delay difference of the two paths to maintain tracking performance.

The error response is the input to the antenna control system that positions the antenna to maintain alignment with the satellite [7, 12]. Like any control system, the error is driven to zero which aligns the main beam's peak with the satellite. The tracking receiver processes the amplitude modulation produced by coupling the difference outputs on the sum pattern to determine the error response. The signal levels received by support terminals vary over a very wide dynamic range as encountered between geosynchronous and polar satellite orbits. Automatic gain control (AGC) circuitry accommodates this large dynamic range requirement for both tracking and data receiver and its operation must be

evaluated. The antenna control system's closed loop operation must evaluate steady state and transient responses as well. This evaluation injects square and triangular waves into the control system loop to evaluate step and ramp responses, respectively. The control system response is also evaluated by tracking a satellite signal and commanding a pointing offset in the azimuth plane, for example. The commanded offset is turned off, allowing the control system to realign the antenna with the satellite. The trajectory back to the satellite indicated by the positioner encoders should remain in the azimuth plane and deviations from the azimuth plane indicate uncompensated group delay in the monopulse circuitry. The process is then repeated in the elevation plane. Such measurements are commonly referred to as "snap-on" tests.

One problem results when systems have a significant margin in received power levels. The antenna can be aligned with a sidelobe rather than the desired main antenna beam. Two open-loop techniques [13, 14] have been devised to verify main beam antenna alignment. One technique offsets the antenna's angular position to measure the received signal's lobe width. The angular width of the main beam is roughly twice that of the sidelobes, allowing the desired verification. The second technique again uses commanded angular offsets and examines the monopulse error slope. The slope in the sidelobes is much greater than that of the main beam, again allowing the desired verification.

The antenna positioner mechanically aligns the antenna with the satellite alignment. Typically, the positioner is an elevation-over-azimuth design where the elevation axis rotates above the azimuth platform. For systems required to track low-altitude satellites, such a mount requires a rapid azimuth rotation to track low-altitude satellites at high-elevation angles, commonly referred to as the keyhole problem. This problem can be addressed by using a third axis to offset the elevation axis. For low-altitude orbits, the typical positioner rates are 15^{o}/sec and 15^{o}/sec^2. During site installation, the antenna positioner must be properly leveled and then indexed to true north to properly register the antenna in inertial space. Typically, the positioner base is marked often with a bead of welding so that its position remains known.

2.4.3 Antenna Site Acceptance Evaluation

Site acceptance tests evaluate several antenna requirements. The antenna's RF performance is evaluated by comparison between the radio source measurements conducted at the vendor and site locations. The antenna tracking capability is first evaluated by demonstrating the antenna can open loop point accurately at several different satellites using their ephemeris values to demonstrate the installed antenna is accurately registered in inertial space coordinates. The positioner's range of motion and compliance with angular rates are also evaluated. The positioner incorporates azimuth and elevation hard and soft stops to avoid damaging the antenna that requires demonstration. The

positioner secures the antenna by applying brakes that are evaluated. Smaller antennas without radomes are equipped with stow pins that secure the antenna in a zenith position when high winds that could damage the antenna are anticipated. The closed-loop tracking response is evaluated by measuring step and ramp responses and conducting snap-on tests using several different satellites. Other RF tests evaluate the antenna's transmission path between the terminal's transmitter, typically located at ground level, and the transmitter's diplexer port. The transmission path uses coaxial cable, waveguide section, rotary joints to allow azimuth and elevation rotations, and flexible waveguide sections to compensate CTE differences. Swept frequency measurements using network analyzer instrumentation determine the frequency response, return loss, and insertion loss end-to-end over the range of positioner motion. The power handling capability is likewise verified.

Network analyzer instrumentation can measure both the insertion loss through the device being tested as well as the voltage reflection coefficient R from the device ports. Three parameters commonly characterize the reflected components. The first is mismatch loss that equals -10 log(1 - $|R|^2$), the second is return loss that equals -20 log($|R|$) [the – sign is used since loss has negative dB values since it's bounded by 1] and the third is voltage standing wave ratio (VSWR) equal to (1 + $|R|$)/(1- $|R|$) that is the peak and minimum voltage values when the direct and reflected components are in-phase and out-of-phase, respectively. Example values are in Figure 2-6.

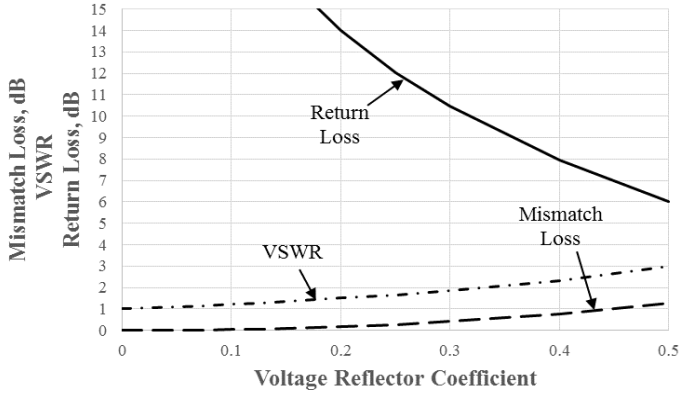
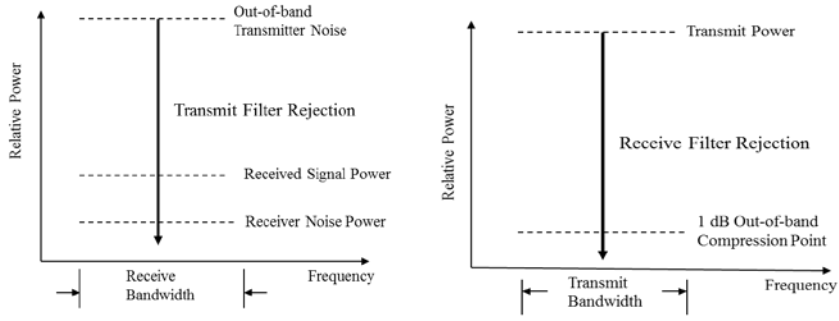


Figure 2-6. Reflection measures.

2.5 Diplexer

A diplexer connects the antenna to the receiver and transmitter and has the necessary filtering (Figure 2-7) so that the transmitter does not interfere with the

receiver. When more than one transmitter and/or receiver frequencies are required, the diplexer is referred to as a multiplexer.



(a) Transmitter filter isolation (b) Receive filter isolation

Figure 2-7. Diplexer isolation requirements.

The filtering transmitter passes transmitter signals with low loss and minimal distortion and rejects out-of-band transmitter spectrum within the receiver bandwidth to a level below the receiver noise power. The required rejection in the receiver's bandwidth depends on the transmitter power level and its out-of-band spectral components. Similarly the receiver filtering passes received signal with low loss and minimal distortion and reduces the in-band transmitter spectral components that are outside the receiver's bandwidth to a level that does not exceed the 1 dB compression point receiver analog electronics at the transmitter frequency. The analog electronics include the LNA that is down converted and filtered with more selectivity at the intermediate frequency (IF) level.

The diplexer is evaluated using network analyzer measurements to characterize the respective in-band and out-of-band characteristics providing the insertion loss, return loss, and amplitude flatness and phase linearity between their respective transmit and receive ports and the antenna port. In addition, the isolation measured between transmit and receive diplexer ports assures compliance with isolation requirements. In addition, the RF power handling capability of the transmitter to antenna path is also addressed, and any evidence of breakdown, passive intermodulation (PIM), or excessive thermal rise at the maximum transmitter output power are investigated. During system integration, diplexer isolation is validated by comparing the receiver's system level with and without the transmitter on. The difference in the two noise levels must be less than 0.1 dB to ensure the diplexer isolation is not degraded by leakage from system connections. This measurement uses spectrum analyzer instrumentation after ensuring the spectrum analyzer's noise does not contribute to the system noise level and averaging techniques reduce the noise variance. This

measurement approach and the allowable noise level increase are commonly used in system specifications.

2.6 Receiver

The receiver functional diagram in Figure 2-8 has an analog front end with a bandpass filter that provides part of the diplexing, an LNA that principally establishes the receiver noise temperature, downconverters to IF frequencies, IF amplifiers incorporating AGC circuitry to maintain the analog-to-digital converter (A/D) linearity, and an A/D output. Digital technology is commonly used in receiver implementations. The A/D must have sufficient sampling rates, 2.5 times oversampling is common, and sufficient quantization to accommodate the anticipated dynamic range following AGC operation. Digital processing serves several purposes. The AGC setting is dynamically derived from the received A/D signal power to maintain linear A/D input levels by varying the IF gain. The input signals are demodulated and separated into its components that include TT&C data, ranging signals, command authentication, and in some cases, mission data. The demodulated signal components are then disseminated to the desired system functions and retained in storage. The data is time-tagged with an inter-range instrumentation group (IRIG) code [15]. The digital processing circuitry acquires and tracks the input carrier frequency and maintains bit and frame synchronization. This circuitry uses a wide-loop bandwidth for acquisition and a narrow bandwidth to maintain signal and Doppler tracking.

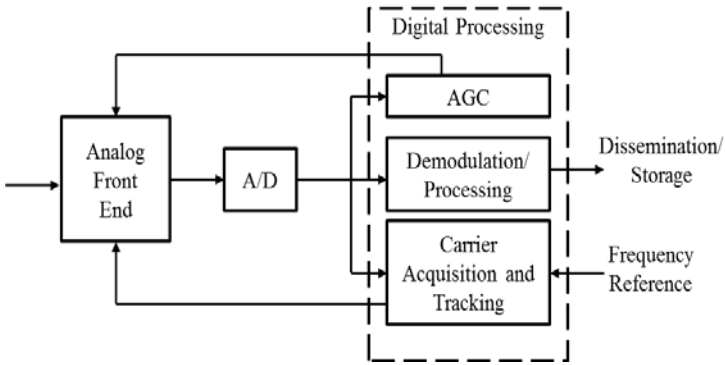


Figure 2-8. Receiver functional diagram.

The receiver characterization evaluates several performance measures: the input noise temperature, the dynamic range, the amplitude and phase flatness, the rejection of out-of-band signals, the signal acquisition and demodulation, demodulation performance, and the interfaces with data processing and storage. The receiver's input noise temperature is comprised of the LNA's noise

temperature and the contributions of the cascaded components from the remaining system. The resulting receiver noise temperature equals

$$T_{Rec} = T_{LNA} + \sum T_i/G_{i\text{-}1} \tag{5}$$

The noise contributions of the components following the LNA are reduced by the insertion gain between the LNA input and the input of the component being addressed. The gain distribution is selected so that the LNA is the dominant noise temperature contribution. Very large LNA gains reduce the cascaded noise contributions, but limit the receiver's dynamic range and increase the susceptibility to high-level interfering signals that are filtered with increased IF selectivity. Thus, a tradeoff exists between low noise and dynamic range that is examined for each application. Large dynamic range capabilities are commonly achieved by AGC circuitry that reduce receiver gain to avoid nonlinear operation but impact the input noise temperature. However, high signal levels that activate the AGC can increase system noise temperature and reduce the output signal-to-noise (S/N) values.

Analog electronic components are commonly characterized by their gain, operating bandwidth, input noise figure, amplitude flatness and phase linearity, and 1 dB compression point at the output which quantifies their linearity limitations using network analyzers. The end-to-end network analyzer response following Figure 2-7 also quantifies the IF filter selectivity and its ability to suppress out-of-band interference. Component noise characteristics are commonly expressed as noise figure whose conversion to noise temperature is given by

$$T = 290(F - 1) \tag{6}$$

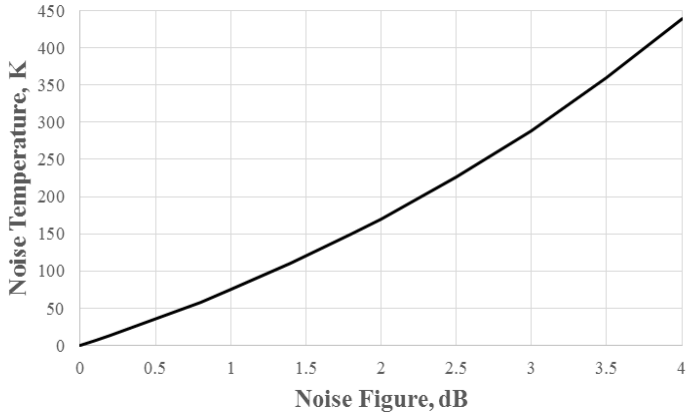


Figure 2-9. Noise figure conversion to noise temperature.

Two alternatives exist to measure system and component noise performance. The first capitalizes on existing commercial noise figure meters and is commonly used to verify component level values. The second is a hot/cold *Y* factor technique commonly used to measure the overall receiver noise temperature. The LNA input is terminated at a cold temperature (77 K for liquid nitrogen is a common choice) providing a T_C reference temperature followed by an ambient temperature termination (290 K) providing a T_H reference temperature. The output noise power is measured for both terminations yielding noise outputs proportional to $T_H + T_{Rec}$ and $T_C + T_{Rec}$ for the hot and cold references, respectively. The noise/power ratio of hot to cold forms the Y_{Rec} factor and the receiver noise temperature equals:

$$T_{Rec} = (T_H - Y_{Rec}\, T_C)/(Y_{Rec} - 1) \tag{7}$$

The receiver dynamic range extends from the minimum required input signal power to input levels that would produce nonlinear analog or digital responses. The minimum input signal power depends on the receiver functions and differ between carrier frequency acquisition and tracking, ranging, and data demodulation. The input carrier power for acquisition and tracking is measured by injecting the carrier signal and determining the minimum value required by the acquisition circuitry. Ranging signals use pseudorandom sequences and the minimum ranging signal level is determined to obtain reliable ranging performance. Received data signal performance is evaluated using bit error rate (BER) that quantify the minimum signal level to achieve a specified error rate, e.g., 10^{-6} that means on the average, one bit in a million will be in error. The input signal level is varied to determine error rates that are $\pm$10 times the specified value, e.g., 10^{-5} and 10^{-7}, so that the slope of the BER curve can be obtained. The BER measurements require sampling at least three to four times the number of bits associated with the BER requirements, e.g., if the BER is 10^{-6}, three to four million bits must be examined to produce statistically reliable results. The comparison between measured and theoretical BER values at the specified BER value quantifies the implementation loss that evaluates receiver relative to an ideal receiver.

2.7 Transmitter

The transmitter is required to modulate the input signal collection, upconvert it to the uplink frequency, and amplify it to the transmitter's selected operating point. The functional diagram in Figure 2-10 illustrates digital technology is typically used to modulate and combine the input data signals into their transmitted format. These digital basebands are converted to the analog domain by a D/A (digital-to-analog converter), upconverted to the transmitted frequency, and amplified to the required output power level.

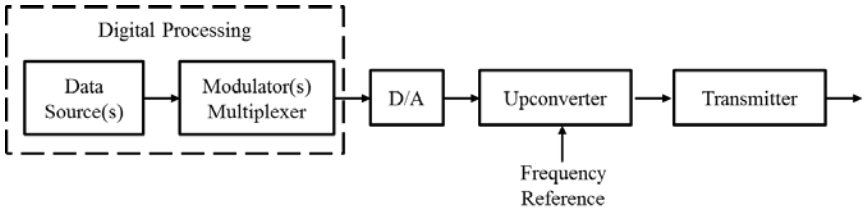


Figure 2-10. Transmitter functional diagram.

The digital processing portion of the transmitter requires evaluation of the commanding, ranging and data transmission capabilities and their combining into the uplink data stream. The individual signals are modulated to form the components of the data streams and combined by a multiplexer to form the baseband uplink data stream. The command signals are required to accommodate the modulation alternatives and data rates and their ability to respond to the command authentication signals received on the downlink must be verified. The ranging codes are generated digitally and at baseband are processed to verify their correctness. The transmitted signals are time-tagged using an IRIG code. Like the receiver, BER measurements are performed using test signal generators to evaluate modulation performance.

After baseband modulation and multiplexing, the signals are converted to the analog domain, upconverted, and amplified to obtain the desired power output. The output signal level is required to vary to satisfy link closure requirements; the required dynamic range over which the transmitted power level can be varied is demonstrated. The linearity at the maximum power output is evaluated quantifying the 1 dB compression point at the output and the AM/AM and AM/PM conversions and harmonic and spurious output levels are established. In operation, the transmitter is backed off from its saturated output to obtain acceptable linearity. The amplitude and phase flatness of the analog circuitry is measured from the D/A output to the transmitter output over the specified dynamic range. Generally, the transmitter output is connected to a switch to route the signals to either the diplexer input or to a test coupler and dummy load so that the transmitter can be tested without signals being radiated.

At this writing, much progress has been made in wide bandgap power devices, such as gallium nitride (GaN) technology that advantageously offer increased output and efficiency. The benefits for terminal transmitters include reduced prime power consumption and cooling requirements. Such transmitters are sufficiently compact to allow locating them on the antenna itself, eliminating the need for waveguide, cabling, and rotary joints required for transmitters located at ground level. The transmission loss of these connecting components is also eliminated and reduces the required transmitter power to satisfy the system ERP requirements. This technology affords potential opportunities that have

operational attractiveness. The progress of this technology should be followed and, as appropriate, serious consideration is recommended for its application to existing and future system designs.

2.8 BITE Capabilities

BITE is an essential element in ground terminals to restore service in the event of terminal malfunction in a timely manner. This importance is increasing with today's trend to remote terminal operation that requires commandable BITE capabilities. The requirement to maintain availability includes financial incentives and needs perceptive fault isolation capabilities, adequate redundancy, and attention to sparing at the site to avoid long delays in restoring service. Such sparing might include replacement electric motors for the positioner drives which have a relatively long delivery from vendors. Attention during terminal development is required to have an adequate number of test points and instrumentation capabilities. Test couplers are required at the LNA input and transmitter output to measure end-to-end responses. The transmitter test coupler is generally located on the switch path to the dummy load so that the testing can be accomplished without radiating through the antenna. Other test points connected to power detectors verify the appropriate power levels are maintained throughout the system and the appropriate AGC settings are commanded. These testing requirements for the analog circuitry can be accomplished by injecting continuous wave tones (CW). Other test requirements necessitate signal generators to replicate operational signals and BER test capabilities to determine proper operation of the uplink and downlink terminal processing to obtain an end-to-end terminal evaluation. These test generators must evaluate all signal components used by the terminal. Generally, a test transponder is constructed to perform end-to-end loopback testing from the LNA input to the transmitter output. The functional diagram in Figure 2-11 illustrates the BITE configuration.

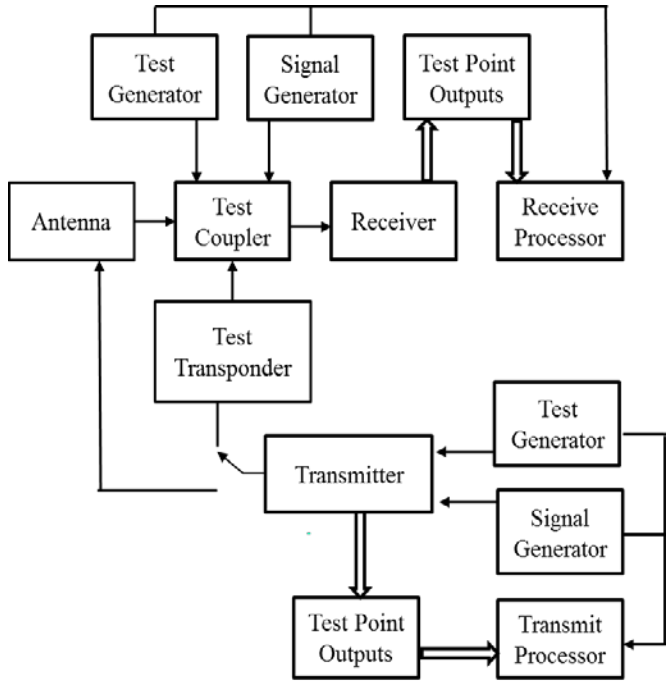


Figure 2-11. BITE functional diagram.

Potential antenna malfunctions require other test capabilities. Typically, the antenna control system stores radio source ephemeris to allow both antenna pointing commands and G/T measurements using radio source techniques that may be remotely commanded to evaluate uplink performance. Downlink performance can be verified by monitoring transmitter power levels at a test coupler using power detectors and test signal generator responses.

2.9 Safety Requirements

Several safety requirements are imposed on terminals. Interlocks are required on power supply enclosures of transmitters to prevent contact with high voltages. Antenna positioners are normally commanded from the site facility, and when necessary may also be commanded by local control at the antenna locations. When local control is used, an interlock precludes unintentional commanding at the site facility. In addition, “kill switches” are prominently located on the antenna to apply the positioner brakes to prevent movement. Because large structures are involved, Occupational Safety and Health Administration (OHSA)-approved safety harnesses are required to obtain access to portions of the antenna that are not available from railing-protected stairs and service platforms.

A second concern is RF radiation hazard when personnel can be exposed to high RF power densities radiated by the antenna. While appropriate power density levels have had much recent debate for many applications, current thinking [16] distinguishes personnel in a controlled environment having a recognized safety program from the general population within the controlled environment. The levels for personnel having a safety program is 100 W/m^2 (10 mW/cm^2) and for the general population is 10 W/m^2 (1 mw/cm^2). Terminals located in foreign countries must comply with their requirements, if more stringent.

Compliance with radiation hazard requirements is generally assessed by using computer codes to determine the antenna's near-field power densities to identify areas and locations having potential vulnerabilities. The analyses are validated by measuring the actual power densities at the identified locations. These measurements are typically performed by replacing the operational transmitter with a low power signal generator to protect measurement personnel and detecting the power density with a suitable probe antenna and a spectrum analyzer. Having assessed and validated the power density levels, areas of concern are clearly marked out and generally, red warning lights are used to indicate the transmitter is operating. Finally, the large antennas used in these applications may also require red aircraft warning lights. These requirements can differ in foreign countries.

2.10 Supporting Software Interfaces

The terminal hardware requires supporting software for scheduling, system configuration and commanding, data processing and dissemination, and data storage, and retrieval illustrated in Figure 2-12. Mission control terminals have additional requirements that are specific to their program applications.

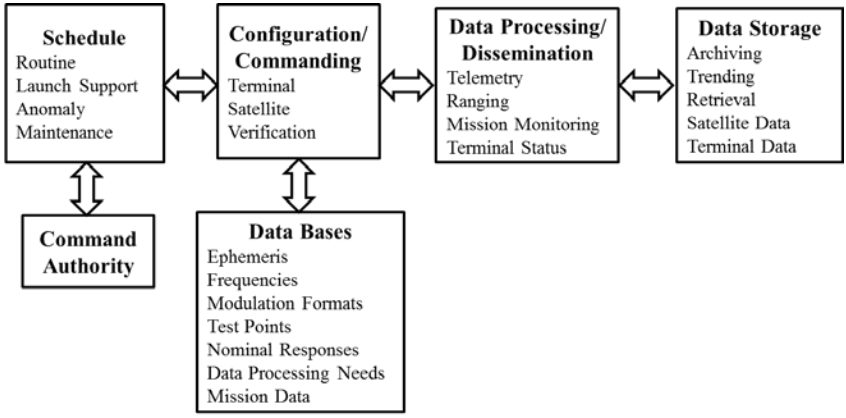


Figure 2-12. Support software needs.

2.10.1 Scheduling

An important software function is the capability to schedule and allocate terminal assets. Some scheduling is routine and is preplanned. Other scheduling for launch support and anomaly resolution must be integrated with more routine scheduling requirements. Generally, anomaly resolution has a significantly higher priority than routine scheduling, but by its nature, the required test time and schedule is difficult to predict *a priori*. Scheduling priorities are dictated by the system's command authority.

Routine scheduling allocates support terminal time between monitoring telemetry outputs that evaluate the satellite's on-orbit health and status and more detailed evaluations performed by mission control terminals that continuously assess payload operations. Such scheduling addresses the satellites to be evaluated, their ephemeris values that require periodic updates to allow terminal access, a listing of both routine satellite telemetry parameters and other satellite test points that can be commanded to support anomaly resolution, and mission support tasking specific to the program application. Support terminal evaluation must use its BITE resources to assure support terminal malfunctions are not mistaken for satellite issues. Routine scheduling must also include allocations for terminal reevaluation and maintenance.

Launch support activities require both satellite and mission control terminals. Satellite support terminals monitor the satellite during its assent to an initial orbital location where the on-orbit tests are conducted. The satellite support terminals command the deployment of solar arrays and other subsystems to configure the satellite to its on-orbit state. During this period, the support terminal also continues to monitor the health and status telemetry, determines the attitude orientation and stability, and helps derive ephemeris values for subsequent testing by mission control terminals. Prior to launch, detailed test plans and schedules for the initial on-orbit period are developed for execution by the mission control terminal. These test plans evaluate the satellite's key performance requirements, measure both primary and redundant paths, and all operating modes to be used by the satellite. For example, the key performance parameters for communication satellites include G/T and ERP values, coverage and polarization characteristics, antenna pointing capabilities, payload characterization measures such as BER that depend on mission applications, and the commanding and resource allocation required by the operational satellite. Evaluations of navigational satellites include the coverage, ERP, code commanding capabilities, and pseudorange performance. Evaluations of remote sensing include sensor performance and its data storage and relay capabilities. The on-orbit test results are carefully archived because these results are the baseline data for the satellite that is referred to over the satellite's lifetime. The schedule for the on-orbit tests requires sufficient contingency to allow further retesting, and other diagnostic tests as needed, to resolve any perceived

performance shortfalls or questionable data. At the completion of this on-orbit testing, the satellite is commanded to relocate to its intended orbital position; a limited number of tests are conducted to assure the satellite's performance is maintained.

Anomaly resolution testing by its nature does not have the luxury of preplanning. Depending on the anomaly, satellite and/or mission control terminals can be tasked in understanding the nature of the anomaly. The testing often requires outputs from commanded telemetry points that normally are not broadcast to better define the anomaly. The satellite's safety and "do no harm" procedures are of utmost importance. Anomaly resolution proceeds by initial tests to better define and verify the anomaly, an examination of alternative reasons for the anomaly, further testing to verify the causes, additional testing to demonstrate alternatives to correct or compensate the anomaly causes, and to define operational performance capability. In addition, a careful examination of the support terminal using BITE resources is required to assure the anomaly is isolated to the satellite. The hypotheses derived from this process identify the anomaly's potential root causes. In some cases, development or engineering model component and/or subsystems are available that provide ground-based opportunities to validate these hypotheses without involving flight hardware. Based on the available data and information, decisions can be made to determine if additional test time and schedule is required and commanding can be initiated to correct or further evaluate the satellite's anomaly; such commanding can include using redundant components.

2.10.2 System Configuration and Commanding

System operation requires configuring the terminal to support the commanded tasking and to properly interface with the data processing, dissemination, and storage capabilities. The correct operation of the configuration and commanding capability is verified by using the BITE assets. Commanding is protected by encryption and authentication techniques.

Databases are necessary to define configuration and commanding information. The ephemeris data for the satellites of interest are needed to command the antenna's positioning. Antenna control systems commonly accept and store two line-element sets of satellites that the terminal could potentially be tasked to service. Attention is required to maintain current ephemeris values in the database. Similarly, the frequency assignments, modulation and demodulation formats, and required ERP levels are necessary to configure the terminal. This database must be verified to insure compatibility with the satellite's transponder. The tasking information includes contact time and duration, the data to be gathered, and any special tasks that differ from routine tasking. For example, if a commanded satellite test point is selected to provide additional data, a database is required that describes the test points that can be commanded and the

necessary commands for access and termination. Database entries related to the commanded test point are obtained from archived data to provide the values anticipated under nominal operating conditions. The necessary data processing, routing, archiving, and dissemination requirements are determined in response to the specific tasking requirements. The planned test events must also be synchronized with IRIG time codes so that the system responses to the commanding can be registered in time.

In addition to the satellite configuration data, another configuration database is also required to implement the terminal's BITE capability. This database includes the test points and their access commands for the terminal, access to a data base indicating appropriate response values, the commands necessary to implement the system operational verification, and those commands that are necessary to evaluate any shortfalls in terminal operation. The BITE capability also validates the terminal configuration has been correctly performed thus, accomplishing configuration verification

Commanding encompasses tasking for both the support terminals and the satellite. Terminals implement the necessary satellite commanding required to perform the identified tasking that requires the appropriate terminal configuration and its verification as well as verification of proper terminal operation. Satellites are commanded to implement the tasking required by both control and mission support terminals and respond initially by command authentication responses and upon reply, carry out the commanding. The satellite's response is received, processed, disseminated, and archived by the support terminals. Other commanding by mission support terminals respond to satellite configuration changes required by satellite operations. For example, communication satellites can require commanding to reposition antennas to satisfy changing coverage needs, to reset transponder gain levels to accommodate traffic demands, and provide user resource assignments in accordance with traffic demands. Likewise, navigational satellites use commanded updates of clock time offsets and ephemeris values to maintain user navigational accuracy. Commanding to select navigational codes and in the future, power levels and spot beam operation will require additional commanding capabilities that is performed by mission-support terminals. Remote sensing satellites are commanded to broadcast stored data. Such commanding is a normal part of satellite operations. Other commanding can provide access to telemetry data that is not normally broadcast during anomaly resolution events, selection of redundant paths if necessary, correction of stationkeeping variations, changes in orbital locations, etc. Other commanding can upload software upgrades and verify their operation. As onboard software becomes more prevalent in future designs, a means of uploading and validating the upgrades will be required. Some commercial communication satellites use their payload mission antennas for such upgrades to achieve the required data rate capabilities. Additional consideration of the needs, requirements, and

implementation of appropriate data links is required to devise an adequate upload capability for other applications.

The role of configuration and commanding for communication satellites becomes significantly more extensive and dynamic to adaptively manage and assign satellite resources to the diverse and dynamic traffic and coverage demands of system users [17, 18]. This expanded role results from more extensive use of digital technology in satellite payloads where channelized architectures flexibly map the uplink signal collection into downlink destinations. A functional diagram of a general satellite transponder in Figure 2-13 includes user access antenna coverage over the satellite's field of view to request and be advised of their allocations, uplink and downlink antennas that typically use multiple beam antenna technology to achieve required system performance and throughput capabilities, demultiplexers to separate uplink signals in their respective coverage areas, and multiplexers to combine downlink signals to their destination coverage areas, where the signal routing and combining is commanded by the system controller and resource allocator. The payload interfaces can include crosslink subsystems to provide global coverage and typically, the architecture includes a gateway where some of the required processing and commanding is performed that also can provide an interface to and from other communication assets. A further alternative is storing traffic that is not time critical in memory for later transmission as a means of averaging traffic demands.

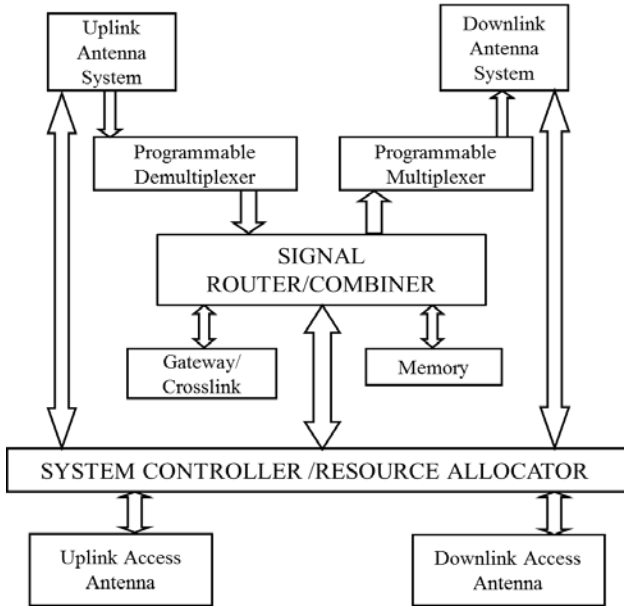


Figure 2-13. Communication satellite payload functional diagram.

This expanded role for communication satellites to dynamically respond to traffic and coverage demands requires an adaptive and dynamic means of implementation. The implementation illustrated in Figure 2-14 receives, acknowledges, and notifies user assignments; determines destination locations and availabilities; adjudicates user priorities after determining commitments and available resources; and commands required satellite configurations. These operations are necessarily referenced to a master clock, because these payloads generally use frequency-hopped spread spectrum waveforms and require synchronization.

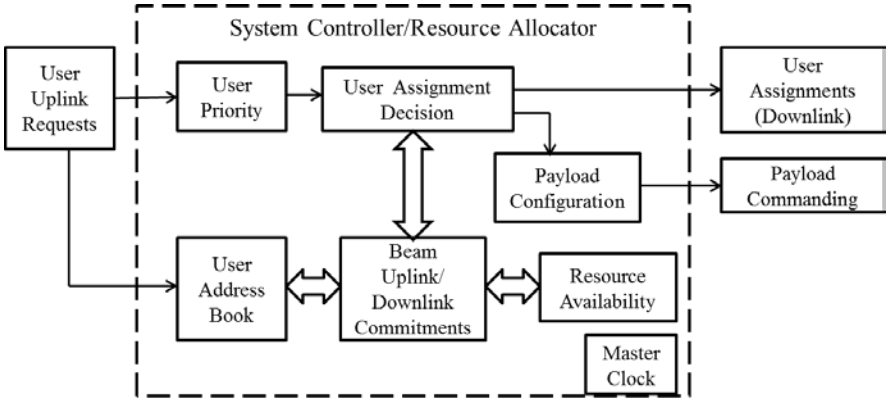


Figure 2-14. Communication satellite resource allocation.

2.10.3 Data Processing and Dissemination

Several data processing and dissemination capabilities are required. Telemetry data is routinely processed to produce ranging data and normal health and status data. The time-tagged, processed ranging data together with the time-tagged, antenna pointing data is distributed to the facility that provides the Kalman filtering of this information and other data to produce ephemeris values. The processed health and status data is routed for retrieval, trending with other data samples, and archived.

More detailed data regarding payload performance is obtained and processed from mission support terminals and depends on the application. Remote sensing applications evaluate image registration, presence of missing pixels, calibration accuracies, and other operational performance measures. Navigational satellites evaluate on-orbit performance by evaluating code components, pseudorange variations, and commanding verification. Further navigational on-orbit monitoring is pursued independently by the intelligence gathering satellite (IGS) community that has an international distribution of sites. Mission support terminals for communication satellites have not only requirements to evaluate

on-orbit performance but more importantly, a more extensive role to allocate service to users, determine traffic limitations, and assess coverage needs.

2.10.4 Data Archiving, Trending, Storage, and Retrieval

The processed data over the satellite's lifetime provides a record of on-orbit capabilities and status that must be maintained. The normal telemetry data is tracked over time; data-trending techniques are required to document performance variations and to identify changes in the data that can anticipate potential performance shortfalls. In addition to the mechanics of trending, established flags to indicate sufficient changes that require additional monitoring and investigation need to be devised and reported. Storage and retrieval is required to assure data can be easily accessed in a clear, reliable manner.

An equally important archiving requirement is storing and maintaining design information and data taken on component, subsystem, and integrated spacecraft levels. Measured development data is required for comparison with on-orbit data. In the event of anomalies, design and development data provides insight that assists in developing hypotheses to explain the anomalies. This data also can assist in determining the effects of out-of-specification conditions, such as voltage variations, so that the system performance under such conditions can be determined.

Finally, the design, development, and calibration data for the support terminals themselves requires effective archiving. In addition to its importance in maintaining terminal operation, such data is equally important to update when sustainment efforts to upgrade terminals are undertaken to satisfy evolved requirements or to replace existing equipment or support software. The documentation must describe changes to the existing design, details of the upgrades, the requirements and evaluation of the upgrades, resulting baseline and calibration data, impacts to BITE capabilities, and baseline data before and after modification.

2.11 References

1. Standardized Interface Specification Between Air Force Satellite Control Network Operations Range Segment and Space Vehicle, SIS-000502D, El Segundo, CA: Space and Missile Systems Center, October 5, 2000.

2. Proceeding of the Apollo Unified S-Band Technical Conference, NASA SP-87, Washington, DC, NASA, July 14 -15, 1965.

3. System Network Users' Guide (SNUG), Revision 10, NASA 450-SNUG, August 2012.

4. Standardized Interface Specification Between the Air Force Satellite Control Network (AFSCN) Network Management Segment (NMS) and Users, SIS-000509E, January 25, 2006.

5. Comparini, Massimo Claudio, Fabio De Tiberis, Rocco Novello, Vincenzo Piloni, Lorenzo Simone, Dario Gelfusa, et al., "Advances in Deep Space Transponder Technology," Proc IEEE 95:10, pp 1994-2008, October 2007.

6. Haskins, Christopher B., and Christopher C. DeBoy, "Deep Space Transceivers- An Innovative Approach to Spacecraft Communications," Proc IEEE 95:10, pp 2009–2018, October 2007.

7. Dybdal, Robert B., Communication Satellite Antennas: System Architecture, Technology, and Evaluation, McGraw-Hill, 2009.

8. Lambert, Kevin M., and Roger C. Rudduck, "Calculation and Verification of Antenna Temperature for Earth-based Reflector Antennas," Radio Science 27:1, pp 23–30, January-February 1992.

9. Baars, W. M., "The Measurement of Large Antennas with Cosmic Radio Sources," IEEE Trans Antennas and Propagation, 21:4, pp. 461–474, July 1973.

10. Dybdal, Robert B., "On G/T Radio Source Measurements," 2000 AMTA Symposium Digest, October 2000.

11. Inoue, Takeo, and Kaitsuka, Toshiyuki, "K-band Tracking System for Domestic Satellite Communication System," IEEE Trans Aerospace and Electronic Systems, vol. AES-17, pp 561–570, July 1981.

12. Dybdal, Robert B., and Denny D. Pidhayny, "Evaluation of Antenna Tracking Systems," 2001 AMTA Symposium Digest, October 2001.

13. Dybdal, Robert B., and Denny D. Pidhayny, "Main Beam Alignment Verification," 2004 IEEE AP-S Symposium Digest, June 20–25, 2004.

14. Dybdal, Robert B., and Denny D. Pidhayny, "Main Beam Alignment Verification for Tracking Antennas," U. S. Patent 6,937,186, issued August 30, 2005, see also, U. S. Patent RE42,472, issued June 21, 2011.

15. "IRIG Serial Time Code Formats," Range Commanders Council Doc. IRIG Standard 200-4, September 2004.

16. IEEE Standard for Safety Levels with Respect to Human Exposure to Radio Frequency Electromagnetic Fields, 32 KHz to 300 GHz, Std C95.1TM-2005.

17. Dybdal, Robert B., “Adaptive Control of Multiple Beam Satellite Transponders,” 1997 IEEE MILCOM Symposium Digest, Monterey CA, November 2–5, 1997.

18. Dybdal, Robert B., “Adaptive Control of Multiple Beam Communication Transponders,” U. S. Patent 6,055,431, issued April 25, 2000.

2.12 Acronyms

A/D	analog to digital
AGC	automatic gain control
BIT	bit error rate
BITE	built-in test equipment
CDA	command and data acquisition
CDR	critical design review
COTS	commercial, off-the-shelf
CTE	coefficient of thermal expansion
CW	Continuous wave
D/A	digital to analog
dB	decibel watts (1 Watt)
DMSP	Defense Meteorological Satellite Program
ERP	effective radiated power
G/T	gain over temperature
GaN	gallium nitride
IF	intermediate frequency
IGS	intelligence gathering satellite
IRIG	inter-range instrumentation group
K	kelvin
LNA	low noise amplifier
MRR	manufacturing readiness review
NOAA	National Oceanic and Atmospheric Administration
OSHA	Occupational Safety and Health Administration
PDR	preliminary design review
PIM	passive intermodulation
RF	radio frequency
S/N	signal-to-noise ratio
SGLS	space-ground link system
SRR	system requirements review
TRD	technical requirements document
TRR	test readiness reviews
TT&C	telemetry, tracking, and control
VSWR	voltage standing wave ratio

Chapter 3
Mobile Ground Element

Chandrakan C. Patel
Systems and Operations Assurance Department
Mission Assurance Subdivision

3.1 Introduction

Mobile ground elements (MGE) have basic requirements and elements comprised of key components and functions. The major difference between fixed and mobile satellites communications depends on their application and requirements. A fixed ground terminal includes terminals on the ground that are in fixed locations; a mobile ground terminal (MGT) can be transportable as part of a MGE. An MGE includes the terminals and system equipment which are operated in fixed locations but can be moved (transported) from one location to another via vehicle, ship, or airplane. A mobile ground system includes terminals and system equipment that may be operated while in motion (mobile) such as while on a vehicle, a ship, or even an airplane.

In today's market, the demand for MGEs is growing rapidly and the distinction between the fixed and mobile is becoming blurred as each is now serving the other's traditional markets. The terminal design is closely tied to the user requirements for the intended end service and application that drive the performance and design requirements for operation. MGTs can use several frequency bands, but has historically relied on L-band and S-band frequencies which facilitate terminal designs. As technology changes rapidly, more and more MGT applications are moving to higher frequencies, such as Ka-band, as capacity demands increase.

A MGE consists of primarily the antenna, RF equipment, and equipment shelters. All these system components are mounted on a trailer, with or without wheels, which can be deployed quickly. Depending upon mission requirements, a radome may also be used. The system components use either commercial-off-the shelf (COTS) or custom products. Some of the COTS products such as ISO containers require some means of transport.

3.2 Background on Ground Systems

A ground station, from the perspective of transmit and receive functionality, has two main purposes and three main functions. The main purposes are to transmit data to and receive data from a satellite. A basic ground system does telemetry, tracking, and commanding. These functions correspond to data and delivery, navigation planning and analysis, and mission control. Ground-system elements

address three main functions performed and monitored by the human machine interface: telemetry, tracking, and command or control (TT&C).

Telemetry typically involves acquisition and processing engineering or payload telemetry from spacecraft to ground. Engineering telemetry provides condition and status of the spacecraft, such as its health, temperature, and the voltage of spacecraft batteries. Payload telemetry provides data pertaining to the spacecraft mission.

Tracking typically acquires information on spacecraft location and follows the trajectory through space. Generally, tracking consists of gathering data on the satellite location and using the data to control ground antenna position. Tracking information is also used for mission data processing. Tracking information typically consists of ranging and Doppler (range-rate) information along with antenna pointing angles.

Command or control applies to the spacecraft by analyzing the engineering telemetry up-linking; transmitting data or down-linking; and receiving data from ground. Each activity requires separate tasks and equipment within the ground system. Generally, commanding refers to the space vehicle while control refers to changing the ground station configuration. A typical ground system can be organized into three areas: ground stations, control centers, and communication links.

A ground station is an installation on the land comprising all the equipment needed to communicate with a spacecraft. Antenna and TT&C hardware at the ground station transmit and receive signals to/from the spacecraft. The ground station may receive commands from the control center, then modulates, formats, and transmits to the spacecraft. Telemetry received from the spacecraft demodulates, formats, and relays it to the control center (or directly to user). The ground station tracks the spacecraft, providing data to the control center for orbit determination, prediction and control as well as sends the data to various users depending upon applications. The primary purpose of the main control center is to function as the main node to control the ground stations to maintain the communications systems that pass data to and from the spacecraft control centers, and also provide host service for the spacecraft control centers.

Because all ground-system elements may not be co-located, a communication link may be required to transport data between the ground-system elements. Communication links complicate a ground system and increase the cost. The best type of link for a particular signal will depend on the type of the information being transmitted and intended use.

Generally the solutions for communication depend on:

Requirements (for data rates, service quality, etc.) call for voice or data signals, (audio, video, digital)
Distribution is point-to-point or multi-drop, and
Speed, distance, security, or survivability.

A fixed terminal generally has a larger antenna and possibly a higher power amplifier (HPA) thus providing a higher effective instantaneous radiated power (EIRP). The fixed terminal can be scheduled for supports without consideration for having to move and set up the terminal. A MGE can be used for services such as factory compatibility tests, launch range compatibility tests, support from areas not serviced by the fixed terminals, providing backup when the fixed terminals are down for long-term maintenance, or major upgrade.

3.3 Mobile Ground Terminal Description

MGEs are ground segment assets that can be deployed to an operational theater and can have several distinct configurations based on program needs. A MGE can operate as a backup mission control center (MCC) and ground station-terminal combination, replicating key functional capabilities of the primary MCC in case of catastrophic failures. MGEs are typically built as transportable vans capable of being airlifted to a designated installation site. Figure 3-1 represents the ground segment reference architecture that illustrates the MGE as an element of the ground segment.

The MGE includes transportable and mobile terminals to transmit, receive or perform both functions for short duration. Figure 3-2 illustrates the typical MGE major components and shows the transport aspects of the MGE. The MGE generally consists of

(a) an antenna dish with or without radome;
(b) shelter(s) to accommodate the communication equipment, the core TT&C electronics, and the high-power amplifier;
(c) a standalone generator or a connection for a power source with necessary electrical and grounding systems; and
(d) a standalone environmental control unit (ECU) or direct mounted ECU on mobile shelter/container

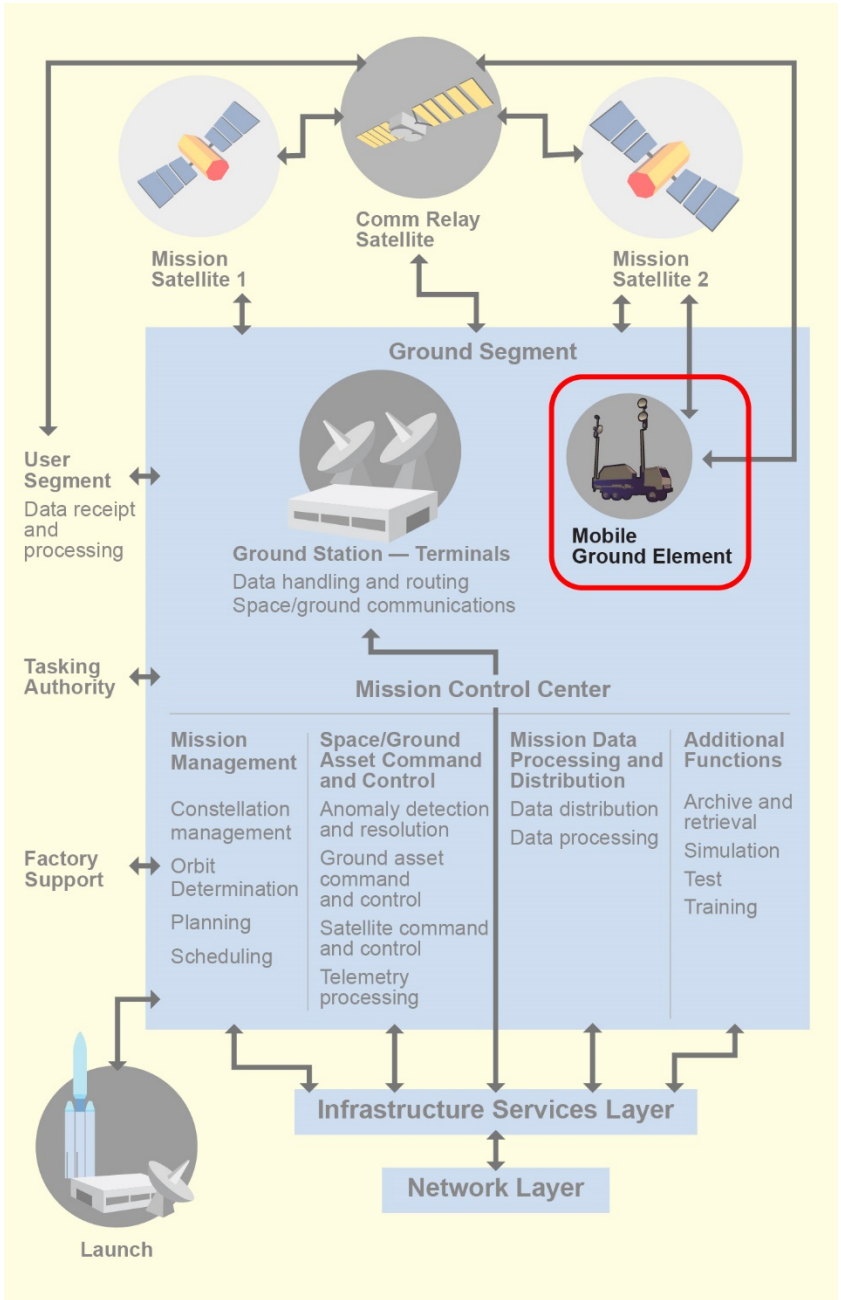


Figure 3-1. Representative ground segment reference architecture with external interfaces.

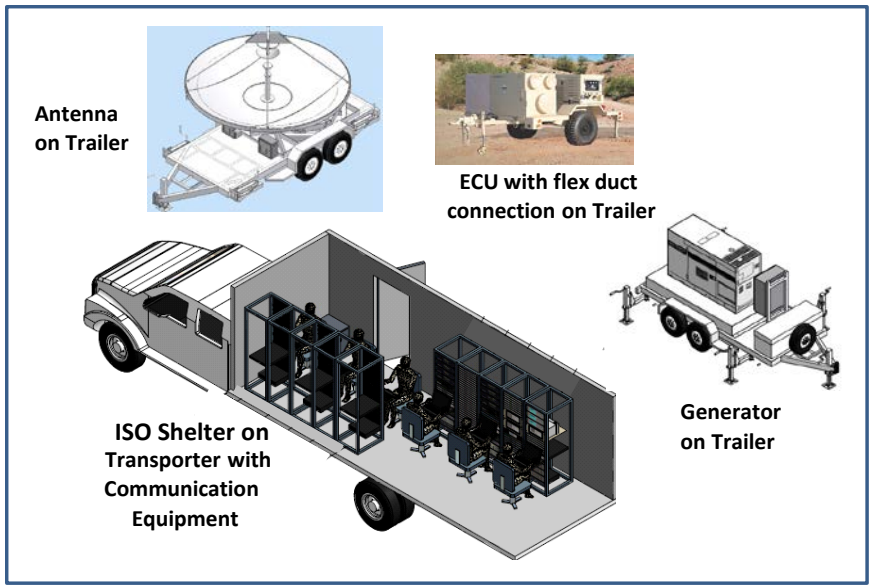


Figure 3-2. Typical MGE major system components.

The antenna will either transmit, receive, or perform both functions for satellite communication. Depending upon mission needs and operations requirements, the MGE site can be manned or unmanned and is part of the normal operations/functions that can be conducted remotely. The system is generally designed to be temporary at a remote location, as the system could be relocated.

The MGE is generally available in various features depending upon user requirements and generally is available in C-, X-, Ku-, L, S and tri-band configuration. The antenna, together with the RF equipment, is mounted on a trailer and either automatically folds into a compact unit without any disassembling, or the system can be disassembled, with the exception of the RF equipment. Figure 3-3 illustrates the typical modules of the MGE required for the data process flow for uplink and downlink from a satellite.

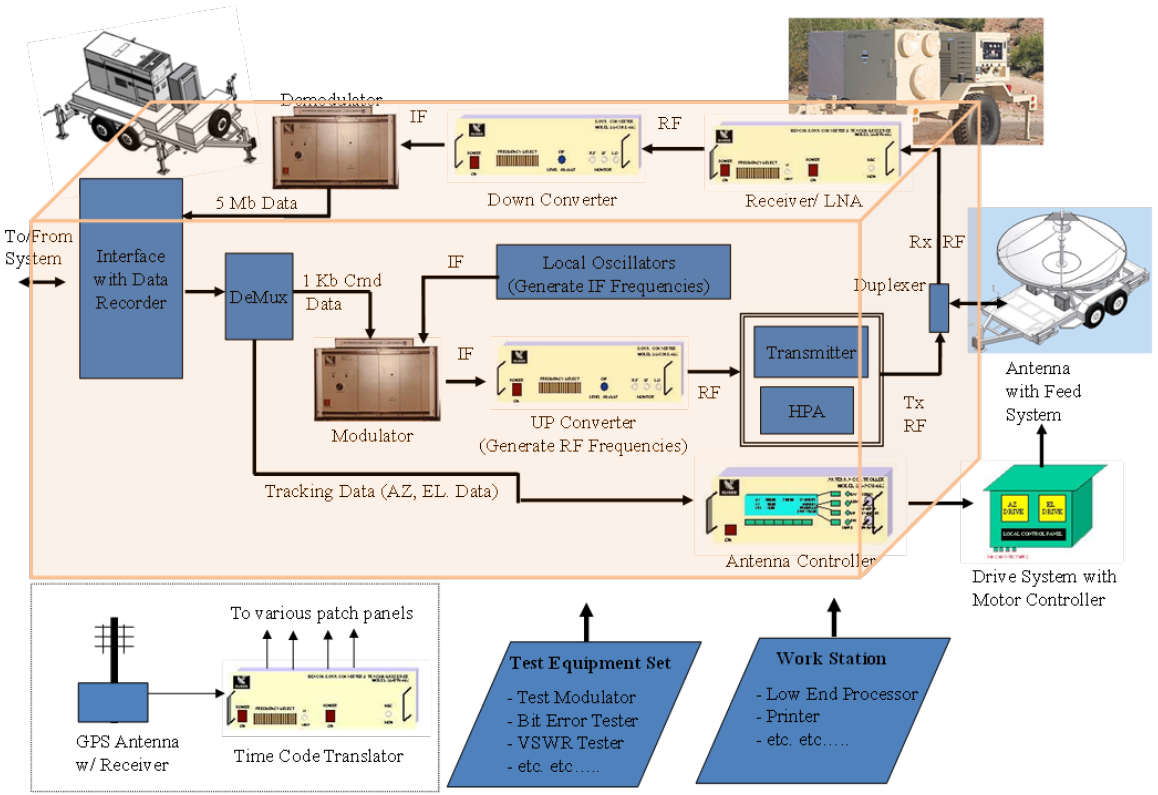


Figure 3-3. Typical modules for data process flow for uplink/downlink from satellite.

3.4 Technical Considerations

Technical requirements depend upon the critical mission needs. A key purpose is to provide a communication link to receive and broadcast information reliably at all times in the out-of-area. Generally, a mobile and transportable satellite ground terminal uses single band configuration L, S, and Ka-band or tri-band (C-, X- and Ku-Band). A MGE is designed for rapid set-up and operation under harsh environmental conditions, and is available in different configurations ranging from a trailer, to a vehicle-mount, to a fly-away configuration in transport boxes.

The site location survey and available access should be evaluated before deployment of the MGE. Primary technical considerations should include:

- User-defined communication requirements
- Independence from the terrestrial infrastructure
- Transportable terminal infrastructure needs – e.g., a pad
- Mobile, transportable terminals ability for deployment to any location
- Ability for rapid or quick deployment of equipment
- Ability for pre-configured terminals
- Inclusion of all required services—data, voice, video
- Robust equipment for multiple deployments
- Easy logistics through air transportable aboard cargo aircraft (i.e., C-130, C-160), train, ship, truck, and ship
- High performance, high-efficiency, multi-frequency antenna systems
- Trailer/container approved or certifiable per commercial or government standards
- Towing capability
- System set up time and ready time for transmission
- Readiness for deployment
- Ability to quickly deploy network wherever need it (e.g. as in-theater network hub)

3.5 Sub-System Components

In addition to the key technical requirements of the MGE, the following key features must also be evaluated in consideration of the mission requirements: the antenna physical features, the antenna performance, the required shelter and containers, the power source/generator, the ECU, safety, siting, environmental regulations, utilities, terrestrial connectivity, and physical security.

3.5.1 Antenna Design

The purpose of the antenna is to either transmit, receive, or both. The antenna design should consider as minimum requirement the following features:

- High performance with rapid interchange between various (C-, X- and Ku-) band feeds based on design and requirements
- Robust and rigid construction (e.g., aluminum)
- Fixed or automated
- Motorized antenna with automatic pointing and satellite tracking
- Government agency certifiable (e.g. by Defense Information System Agency)
- Tri-band and/or single band and frequency converters (traveling wave tube amplifier or solid states)
- Monitoring and control system
- Trailer with pneumatic suspension and level control system
- Compact stowage for air-freighting or other transportable resources

3.5.2 Antenna Performance Factors/Specification

Antenna performance factors can also vary depending on the needs of the user. The following specification parameters are typical requirements that should be evaluated with respect to the specific intended application:

- Antenna diameter
- Elevation range- (generally 5° – 90°)
- Elevation travel rate (e.g. < 0.03°/s fine, < 0.4°/s coarse; 10 degrees/ sec)
- Azimuth range – (generally ±200°)
- Azimuth travel rate (e.g. < 0.03°/s fine, < 0.4°/s coarse; 15 degrees/sec.)
- Polarization adjustment (e.g. ±50° for Ku-band, linear polarization))
- Feed—single band or multi-band interchangeable
- Environmental specification
  - Operational wind speed (depending upon deployment site; e.g. generally > 50 km/h with high wind load option 72 km/h, gusting to 100 km/h)
  - Survival wind speed (stowed) (e.g.180 km/h)
  - Temperature range (e.g. from -32°C to +50°C)
  - Storage ambient temperature (e.g. from 50°C to +70°C)
  - Rain (e.g. Up to 1.8 mm/min.)
  - Humidity: Up to 100%, condensing
  - Solar radiation, (e.g. 1,120 W/m^2)
  - Snow: (e.g. Up to 50 kg/m^2)

- RF Specifications
  - Transmit (Tx) frequency range
  - Receive (Rx) frequency range
  - Transmit (Tx) antenna gain
  - Receive (Rx) antenna gain
  - Transmit (Tx) power travelling wave tube or solid state amplifier
  - Effective instantaneous radiated power (EIRP) rating
  - Gain over temperature (G/T)
  - Noise temperature low noise amplifier (LNA)

3.5.3 Shelter/ ISO Container

The shelter or ISO container is used to install equipment racks which contain communication equipment for command, control and data processing. Generally, the equipment shelter comes with wheels, while the ISO container requires a trailer to transport. The ISO container is a COTS product and comes in standard sizes which can be transported anywhere in the world via ground, sea, or air without any special considerations. Depending upon requirements, more than one container can be used in various configurations. Shelter comes either as COTS product or custom designed with a wheeled configuration so an additional trailer is not required. There are challenges with unique and odd sizes of shelter that are not easily accommodated by transfer via air cargo plane. The construction specifications for the shelter/container depends on the specific mission requirements.

3.5.4 Generator/Power Source

An MGE standalone generator generally provides power. For a generator application, refueling the diesel fuel should be included in the specification. The size of the generator is based on total system power requirements plus the ECU power requirements and other miscellaneous power requirements (i.e., perimeter lights, utility connection). These requirements should be specified in KVA with a voltage and power factor. Proper grounding and a lightening protection system must also be defined. An automatic transfer switch is also usually provided to connect and use the local power as a backup.

3.5.5 Environmental Control Unit (ECU)

The ECU is required to maintain the required temperature for the communications equipment inside the shelter. The capacity (in tons) of the ECU is based on the total heat load of the communication equipment plus the skin loads (based on outside ambient and inside operating conditions) from the shelter/ISO container including solar reflection loads. In addition, if there is a HPA for the antenna, a standalone direct ECU is provided.

3.5.6 Safety

Because any antenna would be a major RF emitting source, a minimum safety distance must be maintained around it. A transmit inhibit zone (TIZ) defines the antenna elevation profile around the antenna below which RF transmission is prohibited. In addition, other safety items such as fire detection and reporting, electrical grounding, a radiation warning system, and lightening protection are considered for MGT.

3.5.7 Siting

Depending on the application of the project, an overall footprint and other interface infrastructure parameters are identified and used to select the site location. The site survey should address the potential exposure to the environmental elements associated with the establishment and maintenance of the MGT.

3.5.8 Environmental Protection Agency (EPA) Requirements

For each selected site an EPA mandate process must be followed. This process is one of the longest lead items in terms of schedule and must be considered far in advance to meet the system design and development.

3.5.9 Utilities

Water (potable and reclaimed water) should be considered in the design, if required.

3.5.10 Terrestrial Connectivity

If the connectivity is required, this should be clearly identified in the required specifications.

3.5.11 Security

Depending on the application of the project, security requirements should be identified. Security should include information security, physical security, as well as personnel security measures such as anti-terrorism force protection.

3.6 Summary

An MGE can be either transportable or a mobile ground system. The demand for MGE is growing rapidly, and the distinction between the fixed and mobile is sometimes blurred as demands for services require available assets that are agile

to accommodate different user needs. The ultimate terminal design is closely tied to the user requirements for the intended end service and to the applications that drive the performance and design requirements for operation.

3.7 Bibliography

Specifications and Standards

Army Regulation 70-38 Research, Development, Test and Evaluation of Materiel for Extreme Climate Conditions, Sept. 15, 1979

MIL-STD 810 G, Test Method Standard, Oct. 31, 2008

Environmental Engineering Considerations and Laboratory Tests

ISO 6346:1995 Freight containers—Coding, identification and marking

ISO 668:2013 Series 1 freight containers—Classification, dimensions and ratings

ISO 1161:1984 Series 1 freight containers—Corner fittings—Specification

ISO 1496-1:2013 Series 1 freight containers—Specification and testing—Part 1: General cargo containers for general purposes

Technical Handbooks

AR 56-4 Distribution of Materiel and Distribution Platform Management Department of Army, Washington DC, March 2, 2007

DOD 4500.9-R Defense Traffic Management Regulation (DTR)

Other

Miguel A. Aguirre, Introduction to space systems: design and synthesis, Springer, Inc. New York, NY, 2013.

Peter Fortescue, Graham Swinerd, John Stark, Spacecraft Systems Engineering, Wiley, September 19, 2011.

Space Mission Engineering: The New SMAD (Space Technology Library, Vol. 28), James R. Wertz (Editor), Microcosm Press; July 29, 2011.

Kenneth Brown, Peter Weiser, eds., Ground Support Systems For Missiles And Space Vehicles Literary Licensing, LLC, May 12, 2012.

Bruce Elbert, The Satellite Communication Ground Segment and Earth Station Handbook, Edition: 2, Artech House, June 30, 2014.

Gerard Maral and Michel Bousquet, Zhili Sun, ed. Satellite Communications Systems: Systems, Techniques and Technology, Wiley, 5th edition. February 1, 2010.

3.8 Acronyms

AZ	azimuth
COTS	commercial off the shelf
DeMux	demultiplex
ECU	environmental control unit
EIRP	effective instantaneous radiated power
EL	elevation
EPA	environmental protection agency
G/T	gain over temperature
HPA	high-power amplifier
ISO	International Standards Organization
KVA	kilovolt amplifier
LNA	low noise amplifier
MCC	mission control center
MGE	mobile ground element
MGT	mobile ground terminal
RF	radio frequency
Rx	receive
TIZ	transmit inhibit zone
TT&C	telemetry, tracking, and command/control
Tx	transmit
VSWR	voltage standing-wave radio

Chapter 4
Mission Control Center

Randall M. Onishi
Advanced Demonstrations
MILSATCOM Division

4.1 Introduction

This chapter provides an overview of the ground segment functions that comprise a typical mission control center. The mission control center (MCC) contains the ground segment functions that are required to execute the program mission. The primary components of a mission control center (shown in Figure 4-1) are (1) Mission Management, (2) Mission Data Processing and Distribution, (3) Space/Ground Asset Command and Control, and (4) Additional Functions. This chapter will provide a high-level description of how each of these components contribute to the operations of a mission control center.

4.2 Key Item Descriptions

Satellite Operations Center (SOC) Facility conducting on-orbit telemetry, tracking and command (TT&C) activities for the programs operational satellites. Activities include, but are not limited to, monitoring satellite status and safety, recovering mission data, maintaining bus and payload capabilities, and maneuvering and stationkeeping the satellite throughout its mission lifetime. The SOC is also referred to as the space/ground asset command and control component of the MCC.

Payload Operations Center (POC) A central facility either remotely or co-located with a MCC where a payload can be independently managed and operated. A POC is most typically used to support hosted payloads. The POC may utilize a separate "out-of-band" radio frequency (RF) communications link with the payload (the main satellite TT&C link is termed "in-band"). The separate RF communications link is usually a high-bandwidth link that is used to transmit the mission data that is collected by the payload.

Mission Control Center (MCC) Responsible for creating the mission plan, scheduling spacecraft resources, and selecting ground resources to meet objectives. The MCC determines the spacecraft's orbit and attitude, sending predicted values to the ground station for tracking acquisition. For complex spacecraft, the MCC may sometimes divide functions into two areas: (1) Spacecraft Operations Center (SOC), which controls the spacecraft's subsystem and processes its data; (2) Payload Operations Center (POC), which controls the spacecraft's payload and analyzes the mission data.

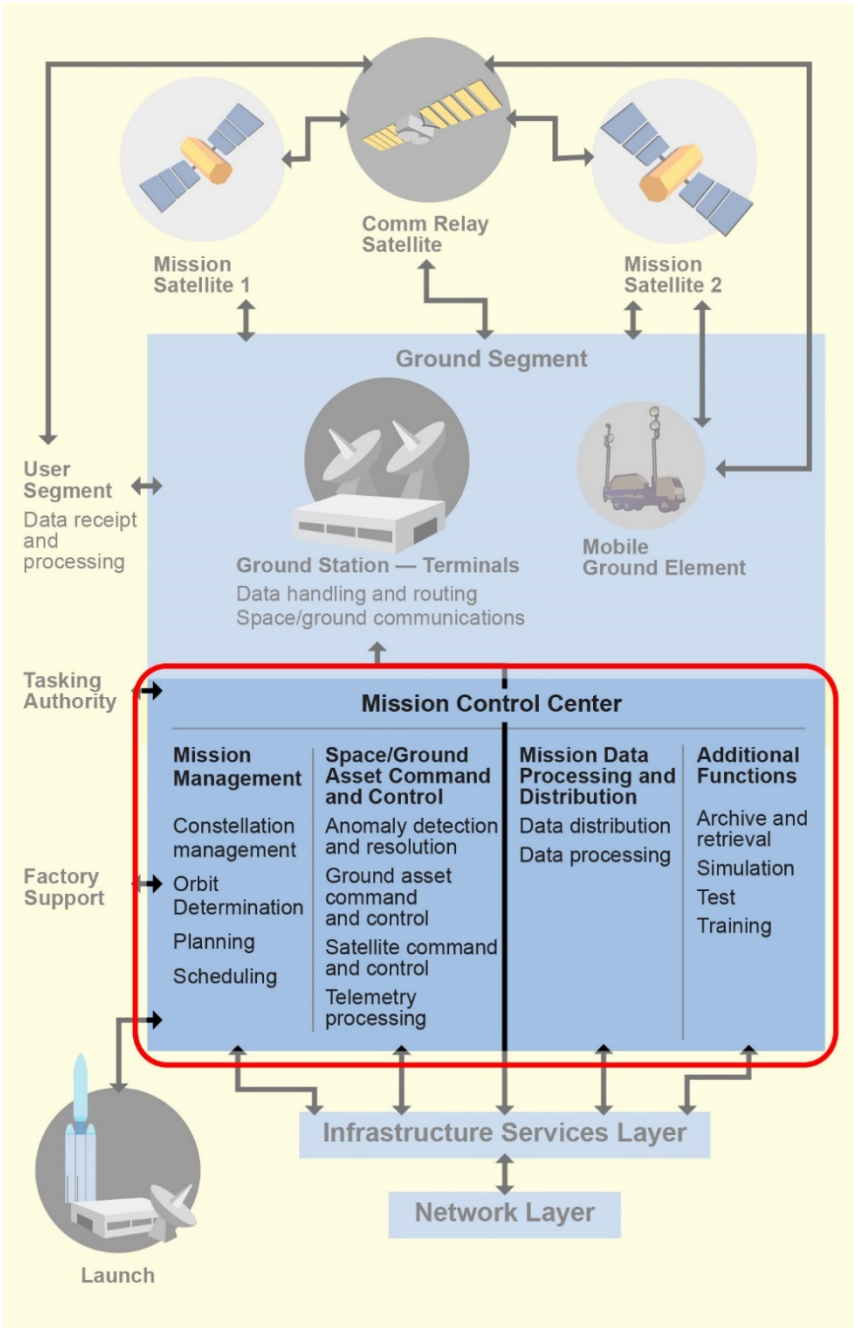


Figure 4-1. Mission control center ground segment components.

MCC Hot Backup Site A "proactive" hot site allows you to keep servers and a live backup site up and running in the event of a disaster. All essential operational databases are replicated at the hot backup site. Basically, you replicate your operational environment at the backup site. This includes an operations floor identical, or nearly identical to the primary operations site. This allows for an immediate transition in case of disaster at your primary site and operators can fill the seats they are familiar with to take over operations. A hot site is a must for mission critical sites.

MCC Warm Backup Site A "preventative" warm site has essential hardware elements of the MCC and operator workstations to be able to take over operations of the mission. Then, if disaster strikes, all you have to do is load your software and data to restore from the latest backups from the primary facility. Getting the warm backup site up and running to take over operations can take many hours.

Anomaly Resolution Support Center A separate room normally at the MCC where the anomaly resolution support team can assemble to work on anomalies and not interfere with normal operations occurring on the operations floor. The anomaly resolution support room has adequate workstations, communications to the operations floor, the space factory, and payload factories for support.

Space/Payload Factories Typically programs will require reachback to the space and payload contractors for support during anomalies, testing, and system updates. The communication network between the factories and the MCC should employ data encryption. Video teleconferencing between the sites should also be considered.

Simulators Satellite bus and/or payload simulators should be procured for installation in the MCC for testing and training purposes. Being able to send new or updated commands to simulators in the MCC will assist in anomaly resolution activities and reduce program risk. Simulators should also exist at the factories.

4.3 Mission Control Center Functional Descriptions

The MCC functions are the essential ground capabilities that are required to operate the mission. This includes mission management for the collection of requests and generation of the mission schedules for both the satellite bus and hosted payloads. Space/ground asset command and control executes the mission schedule which contains the commands for the space assets, the satellite bus and the payloads, and pointing/tracking commands and parameter settings for the ground terminals. Also essential are mission data processing and distribution for missions that require data processing. Not all programs will require mission data processing. Communication relay satellites is an example of a mission that does

not require the data processing function. Additional functions that support training, anomaly resolution, and testing are essential supporting capabilities.

Figure 4-2 is a picture of the space-based infrared system (SBIRS) satellite operations center which is part of the SBIRS MCC. The majority of the MCC functions require operational personnel for mission planning and scheduling, command execution, and monitoring the state-of-health of the satellite and payload.

Courtesy of the U.S. Air Force

Figure 4-2. Space-based infrared system (SBIRS) satellite operations center.

Most programs in the past developed dedicated mission control centers. The concept of "stovepipe" ground systems has been changing as the government searches for ways to reduce ground system development costs. Common ground architectures that can support multiple programs have been developed and are successfully supporting multiple programs. A few examples are the Multi-Mission Satellite Operations Center (MMSOC) Ground System Architecture (GSA) that is employed at Kirtland AFB and Command and Control System-Consolidated (CCS-C) at Schriever AFB that controls the Defense Satellite Communications System (DSCS), Wideband Global SATCOM (WGS) and Advanced Extremely High Frequency (AEHF) satellites. The primary mission control center component that these common systems have been able to share amongst the programs is the command and control function and its associated hardware for transmitting the commands to the space vehicle and receiving

telemetry and ephemeris data. Each program still has their unique mission-planning and data-processing requirements which require program-specific development of the capabilities.

Figure 4-3 shows an overview of the major functions that each of the mission control center components and the interfaces between them. Users' collection requests are sent to mission planning. Requests from all users are de-conflicted and scheduled. Command and control (C2) per the mission schedule converts the commands into the program-defined format for transmission to the space vehicle. Commands are up-linked to the space vehicle via the ground terminal. During contact periods the space vehicle downlinks telemetry which is monitored for state-of-health of the satellite bus and payload. Mission data is downlinked and sent for mission data processing to create data products for distribution.

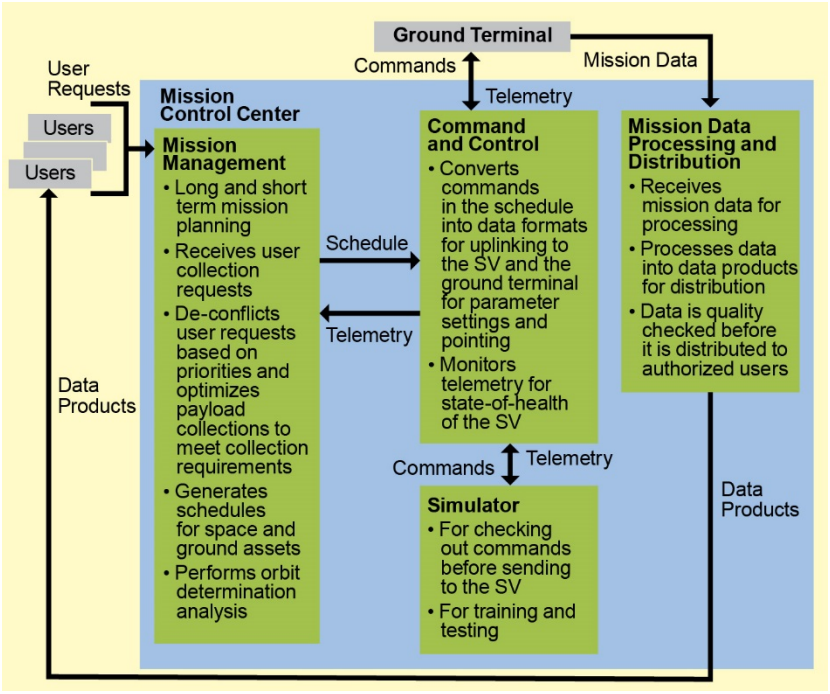


Figure 4-3. Mission control center component functions.

4.3.1 Mission Management

Mission management is responsible for the long- and short-term planning and scheduling for the space, satellite bus and hosted payloads, and the ground assets, including fixed or mobile ground terminals. The mission management

capabilities are usually operated in the same facility as the command and control capabilities. Both are traditionally part of the SOC, which is also referred as the MCC.

The mission management component is responsible for generating a schedule that will be executed by the command and control component. The mission management function can be straightforward for a program that has one space vehicle and a single payload; mission management can be very complex if there are a constellation of space vehicles with multiple payloads that collect data for multiple users.

Mission management for vehicles flying in low Earth orbit (LEO), medium Earth orbit (MEO), and highly elliptical orbit (HEO) are more complex than vehicles that are in the geostationary Earth orbit (GEO) orbit. Based on the requests from users, the mission planners must work with the orbit determination analysts to know the flight paths of the vehicle and ensure collection requests for the specific areas can be satisfied. Other considerations that mission planners must take into account is when the vehicle will be in contact with a ground terminal or relay satellite so that the vehicle can receive commands and downlink mission data and telemetry.

Early in the program the amount of operator workstations that are required to support the mission management function must be identified so that it can be reported to the designers of the operational floor. In addition to the number of operator seats, communication devices must be identified (e.g., unclassified or secure phones).

4.3.2 Space/Ground Asset Command and Control

The space/ground asset command and control function is often referred to as the C2 capability within ground systems. One of the primary functions of C2 is to execute the mission schedule provided by mission management by converting the commands into the program-specified data format for up-linking to the space vehicle. The other primary function is to receive the telemetry from the space vehicle and monitor the state-of-health of the satellite and payload. Most all programs will require the data transmitted between the ground terminal and the space vehicle to be encrypted. Space and ground must coordinate the type of encryption devices that the program will be implementing to ensure compatibility.

Most ground systems development programs in the past employed the C2 system the contractor was most familiar with. Space contractors that build satellites use the same C2 software packages and customize by implementing new commands databases specific to program requirements. The specific tailoring involves the space and payload contractors working with the ground

development team to define commands and telemetry required to meet program requirements. Commands must be able to configure the space assets to maintain orbit specifications and control payloads to collect and transmit the collected data to the ground.

Defining the space-to-ground interfaces that include essential information about all of the critical components on the space vehicle is a critical task. The data in the telemetry stream will be the only data the ground will have access to for monitoring the health and status of the space vehicle and to use during anomaly resolution activities.

The ephemeris data that the orbit determination function uses to determine the actual location of the space vehicle is also included in the low-bandwidth downlinked data. For vehicles in orbit, the orbit determination function provides mission scheduling contact periods. The C2 system will send commands to the appropriate ground terminal regarding where to point and start listening for the space vehicle at a certain time for the schedule contact period.

Air Force Satellite Control Network (AFSCN) ground terminals are low-bandwidth and primarily used for uplinking/downlinking C2 data. Mission data usually requires a high bandwidth downlink capability. Programs requiring a high bandwidth ground terminal procure and install dedicated ground terminals that meet the programs requirements. The dedicated ground terminals must have a terrestrial communications network to the MCC and the mission data processing and distribution facility.

4.3.3 Mission Data Processing and Distribution

Not all programs will require a mission data processing and distribution component. Communication satellites that do not have hosted payloads collecting data requiring processing for exploitation would not require a mission data processing component.

Satellites with sensors that collect infrared (IR), Earth orbit (EO), or radar data would require the mission data processing function. Depending on the amount of data being collected and the timeliness requirements for the processing of the raw data into a useable product drives the type and amount of ground processing hardware required for the program.

The mission data processing and distribution component could be a large development effort. As an example, SBIRS collects vast amount of IR data that is downlinked requiring a large software development effort for the mission data processing center. The mission data processing and distribution component can be co-located with the MCC or in another facility.

In the past, programs have all purchased hardware specifically for their program, now cloud processing is being looked at as a cost savings to programs. The goal of cloud processing is to reduce programs hardware costs. The cloud provides shared and on-demand hardware resources for processing. With the fact that the cloud hardware is a shared resource amongst programs, security is an issue that will need to be fully resolved for the future of cloud computing utilization.

Part of the mission data processing component is also data storage. Programs typically have requirements to store raw data in case better algorithms are developed in the future which can extract more intelligence than the current processing algorithms. Mission data processing will always want to store and archive the products that are disseminated to the users.

4.3.4 Additional Functions

Additional functions are ground system components that provide additional functionality to the mission control center.

4.3.4.1 Simulators

Simulators for both the satellite bus and the payload can reduce risk for the program. Having the simulators at the SOC is desired, but if the program can only afford to have simulators at the factories it is better than having none at all. Simulators can be used prior to launch for testing and training of operational personnel.

Sometime during the program the flight software will be need to be updated. Having a simulator at the SOC or factory allows for the software updates to be tested on the ground before sending it to the actual hardware in space. This can reduce risk of harming any space components with a bad software update. There will be cases where the software update cannot be fully tested with the simulator and can only be tested in space.

4.3.4.2 Archive and Retrieval

The archiving of all decommutated telemetry and ephemeris data for the lifetime of the program is usually a program requirement. The database should be designed so that operators can easily request data from periods of time to be retrieved from the database for analysis. Data from this database will also be used for trending analysis.

4.4 Technical Considerations

The government has many ongoing studies that evaluate future ground architectures (FGA) with a goal of supporting multiple programs. There are a number of mission control center ground segment infrastructure functions that are common in many of the stovepipe systems developed in the past. The FGA will provide those functions and thus save development costs. The programs will only have to develop their mission specific capabilities (e.g., commands, telemetry, and mission data processing). In the future, programs will have to perform trade studies to determine if using FGA is cost effective.

In addition to reducing ground segment development costs by providing a FGA, the government is also studying cloud computing for cost savings. Instead of having each program purchase hardware and build facility space for the hardware, programs would utilize the cloud resources for computing instead.

4.5 Programmatic Considerations

The major programmatic consideration regarding the mission control center for programs will be its location. The program will have to consider facilities that already exist where their program can operate their mission. With the development of MMSOCs and potential other facilities being developed, programs will need to determine a more cost effective approach.

Another major decision when procuring a program regards the requirement for a MCC backup facility. Most programs whose mission is providing critical information to the forces on the ground have a “hot” backup facility. The backup facility not only has the operational hardware that will support the mission, but also has an operations floor that is a replica of the primary site’s operations floor. The reason for this is that programs have found there is a quicker transition when operators can take over operations in a new facility and everything is familiar to them.

Some programs don’t need a backup site, but do have reliability and availability requirements. Sometimes the requirements can be met by having redundant hardware in their one facility. Reliability and availability analysis would be required to determine the hardware and software configurations that would meet the programs requirements.

Other issues that programs need to consider are system backups. Most programs will have a requirement that requires copies of system backups are stored at an off-site secure location in the event of disaster.

4.6 Summary

The MCC contains the critical ground segment components for executing the mission. All of the users' requests are handled by the mission management component that optimizes the mission schedules to meet the program collection requirements. Orbit determination is a key function of mission management that provides the calculated location of the space vehicle so that pointing commands are on target for the requested collections. The C2 component executes the mission schedule by sending commands to the space vehicle and monitors the state-of-health of the satellite bus and payload. All anomalies are handled by the MCC with assistance from the factories when required. Mission data is processed and distributed by the mission data processing and distribution component. Other functions like simulators support the execution of the mission as well as with anomalies.

Where each of the MCC components are implemented will be the decision of the program. Cost and technical risk should be the determining factors. One other major cost component that programs must consider is the cost of the operations and maintenance (O&M) of the program. If the life of the program is projected to be 10 years, then the program O&M costs will include the cost to operate the system and the maintenance of the hardware and software.

Lessons learned from Aerospace Ground Systems Team (GST) Concept Design Center (CDC) studies are the large costs associated with the O&M phase of the program. Often times the costs of the O&M is almost twice the cost of the original development of the program. This fact often surprises the CDC GST customers. They often ask how the O&M costs can be reduced. A couple of cost savings approaches are implementing into a multi-mission C2 center or the use of Cloud processing. Both could save hardware maintenance costs and the multi-mission C2 center could save costs associated with the operations of the mission. Every program will have to perform its own trades and analyses to determine the best solution for the program.

4.7 Bibliography

Los Angeles Air Force Base Fact sheets – Command and Control System Consolidated, posted February 11, 2014, http://www.losangeles.af.mil/library/factsheets/factsheet.asp?id=7852.

Morgan, Tiffany, Multi-Mission Satellite Operations Center Ground System Architecture (MMSOC GSA), Presented at Ground Systems Architecture Workshop, February 29, 2012.

Arnold, David Christopher, Spying from Space: Constructing America's Satellite Command and Control Systems, 2008.

4.8 Acronyms

AEHF	advanced extremely high frequency
AFSCN	Air Force Satellite Control Network
C2	command and control
CCS-C	command and control system-consolidated
CDC	Concept Design Center
DSCS	defense satellite communications system
EO	Earth orbit
FGA	future ground architectures
GEO	geostationary Earth orbit
GSA	ground system architecture
GST	ground systems team
HEO	highly elliptical orbit
IR	infrared
LEO	low Earth orbit
MCC	mission control center
MEO	medium Earth orbit
MMSOC	multi-mission satellite operations center
O&M	operations and maintenance
POC	payload operations center
RF	radio frequency
SATCOM	satellite communication
SBIRS	space-based infrared system
SOC	satellite operations center
TT&C	telemetry, tracking, and command
WGS	wideband global SATCOM

Chapter 5
Mission Management

Erin Y. Carraher
Navigation and Geopositioning Systems Department
System Analysis and Simulation Subdivision
Marc D. DiPrinzio and Robert E. Markin
Mission Analysis and Operations Department
System Analysis and Simulation Subdivision
Marilyn K. Dubas
Software Engineering Subdivision
Computers and Software Division
Justin F. McNeill
JPL and Robotics Programs
Systems and Technology Programs Directorate

5.1 Introduction

Mission management (MM) encompasses planning and scheduling for space and ground assets of a space system architecture. It includes the full cycle of planning and scheduling activities, from long-range planning (of architectures and configurations) through pre-planning of mission activities (for space and ground assets), to real-time reaction to events and circumstances, and on through asset retirement. Performance assessment is also part of total MM functionality. Orbit determination (OD) is a vital capability supporting multiple MM activities such as planning for orbital maneuvers, station keeping and payload and bus operations. Figure 5-1 illustrates MM's placement within the ground segment reference architecture.

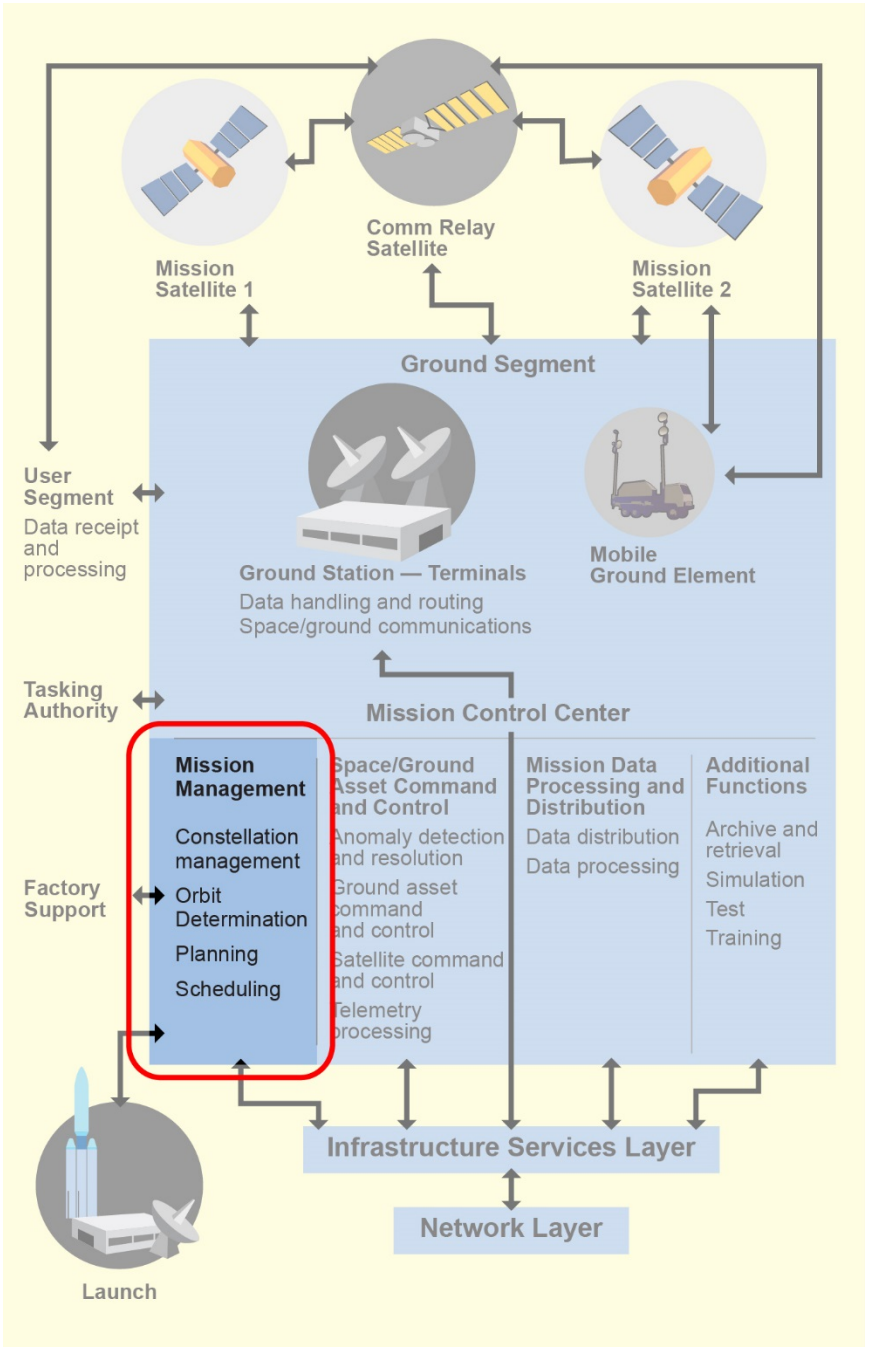


Figure 5-1. Mission management within the ground segment.

Mission management consists of people, processes, plans, software, hardware, and procedures. Mission management supports activities such as space system commissioning, system check-out and testing, system calibration, orbital maneuvers, payload pointing, deployments, anomaly detection and resolution, contingency planning, planning and scheduling operations, and decommissioning. As a central function of MM, an operations team develops a schedule from which the space and ground assets operate. The schedule is based on a collection of user requests, constraints, execution parameters, and predicted or evaluated performance. The process also includes appropriate risk analyses of individual orbit, attitude, vehicle, and payload configurations necessary for the successful execution of the mission objectives.

Mission management nomenclature differs among communities and programs but, in general, planning activities occur first and direct the generation of schedules that drive the execution of space and ground assets. Because the community of government agencies and private entities that operate space systems is broad and diverse, this discussion of generally will be kept at a high level, with emphasis on Earth-orbiting missions. This chapter focuses on the functionality provided by MM subsystems and OD components as well as process and technical considerations as part of the ground system. The chapter also provides some insight to vital planning and scheduling activities necessary in space systems.

5.2 Definitions

Ground asset can be the ground hardware, software, subsystem, or facilities

Space asset Space segment entity associated with a ground system. Usually a satellite (bus plus payloads), but can also refer to hosted payloads.

5.3 Detailed Description

Mission management supports most of the project lifecycle, beginning with the definition of the mission and development of the system's preliminary design, pre-launch planning, continuing with launch and initialization (commissioning), transitioning into daily in-flight activities (operations), and concluding with end-of-life/end-of-mission activities (decommissioning). The nominal phases are illustrated in Figure 5-2.

Figure 5-2. Nominal project lifecycle phases.

MM subsystems are primarily developed to support operations. During other lifecycle phases planning and scheduling activities may be supported with specialized tools which are not part of the ground system's MM subsystem.

5.3.1 Pre-launch

Pre-launch includes various planning activities leading to the launch of a spacecraft or flight system.

Pre-launch planning includes definition of orbits and orbital maneuvers for the period from launch through the initial stages of commissioning. The mission design will define the path to the orbit or trajectory necessary to satisfy the mission objective. Spacecraft are most commonly launched into Earth orbits, but the target body can be any body of interest: the sun, a planet, asteroid, or comet. Pre-launch planning includes analysis to ensure that the spacecraft achieves the mission trajectory or orbit regardless of potential errors (e.g., errors in the separation orbit, errors in the execution of the orbit maneuver, missed burns). Another aspect of the mission design that must be addressed is the propellant budget. After all propellant requirements are met, there must be positive margin for the mission to be successful.

5.3.2 Commissioning

After spacecraft launch and before the declaration of initial operating capability (IOC), the spacecraft generally must pass through a "commissioning" phase in which the spacecraft system is initialized. This can be the highest risk segment of the mission. Mission planning must account for constraints, limitations, and maneuver objectives that are usually very different from those in nominal operations. The consequences of mistakes in this phase are sometimes irreversible and may be catastrophic.

During commissioning, mission planning support may include the following: maneuvering for orbit initialization, momentum management, vehicle deployments, sun acquisition, Earth acquisition, payload checkout and calibration, thruster functionals, thruster conditioning, and spin axis and spin-speed changes. These events are grouped into three segments: maneuvering, bus in-orbit test (IOT), and payload IOT. Initialization of the normal mode (the operations configuration) completes the commissioning phase.

5.3.2.1 Commissioning—Maneuver Phase

The purpose of the maneuver phase is to move the spacecraft from its initial trajectory to a desired operational trajectory. Initial mission planning for the maneuver phase is based on the pre-launch plan and takes into account the actual launch vehicle injection orbit, any propellant expenditures required for

capture and initial acquisition, and updates to the spacecraft health and functionality. Subsequent mission plans are based on the results of previous maneuvers, expected or sudden configurations changes, geometric events (such as eclipse), and predicted thruster performance. Additional mission planning support during the maneuver segment may include launch collision avoidance verification and response, anomaly response, as well as the support of any required ad hoc analysis

The end of the maneuver phase is reached when the current orbit is within the allowed tolerances of the beginning of life (BOL) operational orbit. Sometimes the tolerances may be large enough that the vehicle is drifting intentionally and is expected to maneuver again during the IOT phase and prior to commencing operations.

5.3.2.2 Commissioning—In-Orbit Test for Bus and Payload

Bus IOT often occurs after the maneuvers, although some missions, especially in low-Earth orbit (LEO), perform much of bus IOT prior to maneuvering in order to maintain vehicle health and safety. Payload IOT is almost always after the maneuvering phase. It is during bus IOT that many of the following one-time events occur:

- Solar panel and antenna deployments
- Sun acquisition
- Earth (or central body) acquisition and transition to normal (on-orbit) mode
- Tank isolation, thruster conditioning, transition to blowdown, as applicable

A few other events may or may not occur during bus IOT, including orbit correction and stationkeeping with on-orbit thrusters, momentum dumping, and drift start/stop. Payload IOT, which often requires quiescent attitude and environmental conditions, usually begins once the maneuvering and bus IOT segments are complete.

At some point during IOT, the MM team performs a few other substantial tasks. The propulsion team updates the expected performance characteristics of the on-orbit stationkeeping thrusters. Then the orbit analyst and propulsion teams work together to estimate the remaining propellant on-board the spacecraft, to reallocate unused contingency propellant, to update the propellant budget, and to update the lifetime estimate.

5.3.3 Operations

At the conclusion of commissioning, the space system enters operations. During this phase, the mission's purpose is realized. Many stakeholder communities may be involved in determining what is required of a space system to fulfill its mission. Stakeholders include the flight crews (and the operator community in DOD systems), user communities, and factory support organizations (when such entities support operations). When user/mission needs exceed what the space system can simultaneously accomplish, scheduling and schedule optimization is required to address contention among activities to execute.

Table 5-1 is a top-level summary of MM elements during the operational phase.

Table 5-1. Elements of Planning and Scheduling for Operations

Items	Activities
• Set of activities to schedule o Satellite state of health (SOH), o Payload operations o Product generation o Product distribution o Space-to-ground contacts o Networks o External resources • Planning data/constraints o Ephemeris/attitude o Vehicle/payload models o Ground system constraints o External resource schedules • Displays o System task/activity builder o Time lines o Map projections o Constellation views	• Activity definition management o Tasking receipt and verification o Translation of task definition into system activities o Tasking database management o SOH activities management • Schedule "optimization" o Schedule engine o Optimization function o Manual overrides o What ifs • Schedule approval processes • Schedule promotion process • Translation of schedule to commands • Schedule execution • Ad hoc response • Schedule dissemination • Replanning • Performance evaluation

Satellite bus and payload state of health (housekeeping) activities form the foundation and place constraints upon which, when, and how mission activities can take place. Payload operations and the satellite bus (e.g. maneuvers), as well as ground activities necessary to support those payload operations, are the focus of MM during this phase.

When shared resources (such as the Air Force Satellite Control Network [AFSCN], NASA's Space Network, or the United Space Network [USN]) are utilized for space-to-ground communications, there are formalized processes to arrange for the space-to-ground contact intervals. These procedures take place over multiple weeks and are formalized by the resource providing organization.

Some systems are not designed to be able to support the simultaneous generation of all required output products. In such systems, product generation scheduling is included in MM. This especially may be true in systems where MM supports distribution and exploitation.

Planning and scheduling must take a variety of things into account. MM includes software that models payload and satellite bus actions and constraints. Interaction with space subsystem experts may also be included in MM operational planning. Ground asset capabilities and availability also may place constraints on what and when activities can be scheduled. Estimates of ephemeris and payload/bus attitude are important contributors to planning and scheduling.

5.3.3.1 Tasking

There are numerous steps in the process of planning and scheduling mission operations. The process begins by defining what to do. For several space mission categories, this involves input from external stakeholders and users. In such space systems, the MM subsystem receives requests ("tasking") for mission activities. In missions such as remote sensing (which produce products for users) and communications (which provides services to users), a tasking infrastructure may exist to "formally" identify the needs and/or desires of the using communities. Such infrastructures are usually external to the space system they "task" and often support tasking of multiple space systems. Verification and error checking of the requests are included in the process of storage and cataloging and/or organizing the received requests. Priority is usually associated with requests. The tasking infrastructure, when it exists, is responsible for establishing priority for requests.

Tasking must be translated into system-executable activities prior to the scheduling step. This is often done prior to planning. Even very specific requests are generally not at the level necessary to provide commands to a space asset. MM subsystems manage databases for the raw requests and for the space system unique translation. MM subsystems provide operators database management tools. The "activities database" also includes space asset state-of-health and housekeeping activities. Such activities are identified by flight teams, space asset subsystem experts and space segment factories. Long-term plans and schedule components for routine maintenance activities are often generated and retained in the databases.

Space assets and sometimes ground assets execute based on schedules generated via MM. The schedules include mission, supporting space asset actions, and ground activities when applicable. Schedules are generated for distinct periods of time. Often high-level plans are developed for activities days or weeks away. Such plans are refined as the execution period gets closer and eventually turned into schedules. It is very common for the operational schedule to be generated for a day's worth of activities and to be updated daily. Schedules for the next period of operations overlap in a seamless fashion with the schedule from the last period.

5.3.3.2 Scheduling

Schedule generation is an end goal of operational MM. Scheduling is a resource-allocation discipline. User requests (at the single-asset level and the constellation level) may exceed what can simultaneously be accomplished. State of health (SOH) activities may be required at specific times and therefore have precedence over mission activities that cannot be done in concert with SOH activities. In other cases there is more flexibility as to when SOH activities can be done. Under those circumstances SOH and mission activities compete for schedule placement.

When systems are oversubscribed (users want more than can be accomplished) approaches to schedule optimization are required. In many systems priority of individual tasks is the driver. In such cases the highest-priority tasks are scheduled and lower-priority activities are fit in around them. Other approaches attempt to maximize the number of objectives that are accomplished. MM operators may have options as to how they want the optimization function to make decisions. However, system requirements may dictate the approach to optimization. Many space systems include more than one space asset and schedule generations optimize at the constellation level. Automated scheduling of required maintenance activities such as calibrations may be part of the scheduling rules included in the scheduling engine.

"What if" capabilities are supported by some MM subsystems. Operator selections for which tasks to include, modifiable features of tasks, assets to consider, time period, optimization approach, etc. will affect the schedule generated by the scheduling software. Operators can consider and compare the outputs in these "what-if" scenarios and make the final selection. Most scheduling systems allow operators to manually override or alter a generated schedule.

Because the schedule controls what the space (and ground when applicable) assets do, approval of the schedule for the next period of operations is often a formal process which may involve input from multiple stakeholders. In some systems once the schedule has been approved, a representation of its content is

disseminated to the stakeholders. For this discussion, the term "schedule promotion" refers to the processes by which the schedule approved for the next period actually becomes the operational schedule. Subsequent operations and execution follow that new schedule. Commands for the space assets are generated. The command sequences are constraint-checked to assure that the space assets will not be forced outside of allowable hardware or software limits. The responsibility for the actual command generation may reside with space/ground asset command and control, not with the MM subsystem. Space/ground asset command and control is responsible for sending the commands to the space assets. The interactions between the MM subsystem and the space/ground command and control subsystem are unique to each space system.

As the space asset executes scheduled activities, engineering and mission data are provided in telemetry that is routed to the MM subsystem and serves as an important input. MM tracks actual execution versus what was scheduled. Such comparisons are utilized in performance evaluation. Deviations from commanded schedules may initiate recovery actions such as anomaly responses or rescheduling. When the operations team detects an anomaly or off-nominal operations of a certain criticality, the team will then execute predefined steps to first ensure the safety of the spacecraft and payload. The team will generate an interim mission schedule to operate the spacecraft safely until the anomaly is defined, understood, and either resolved or mitigated.

Some systems are required to respond to "urgent" requests or self-identified situations and make changes in what is to be accomplished. In such systems, MM "quickly" reschedules. Sometimes this results in a new promoted schedule taking over. But it can merely be an override of the promoted schedule for a limited period of time. The input for such near-real-time changes may be internally generated or come from stakeholder input to flight crews (when allowed) or from the tasking infrastructure or a system of systems MM (when applicable).

Some systems are required to periodically disseminate information on what was actually executed by the space system. This "as executed" schedule information lets stakeholders know that their requests were executed.

5.3.3.3 Performance Assessment

Evaluation of space system performance includes the evaluation of schedule execution rather than "tasks to perform." Such comparisons can be utilized to improve the scheduling engine performance. It may also influence the tasking infrastructure when such exists. Performance evaluation is often conducted by entities external to the space systems. MM subsystems may not have

requirements to track scheduling performance. When MM does have such requirements then additional software and tools are provided for analysts.

5.3.3.4 Timelines

MM activities address timelines anywhere from years to seconds. Figure 5-3 is a summary of the timeline which operational MM supports.

The placement of constellation components is planned from months to years in advance (including significant adjustments to orbital locations). Planning for adjustment and upgrades to ground facilities and infrastructure are also done on this time scale. Ground maintenance activities that impact operations are addressed months to weeks out.

As mentioned previously, if a space system utilizes external resources such as those from the AFSCN or a commercial infrastructure provider such as USN, planning and coordination for such services starts weeks prior to the utilization. Planning for scheduled activities often begins days prior to when the schedule will be promoted. Planning to support ground processing, exploitation, and dissemination may be required days prior to the execution. This is especially true when contention for resources is an issue. “What if” scheduling analysis may be done days prior to the generation of the final schedule. Planning activities, including generation of preliminary/potential schedules, may include direct interaction between MM operators and the tasking infrastructure and/or users to work through identified conflicts.

Schedules for the next period of operations are generated and finalized hours prior to the schedule start time. Some systems are required to incorporate late input to their schedules and reschedule up to minutes prior to start time. And, as mentioned earlier, some MM subsystems are required to alter schedules very quickly, even as the schedules are being executed.

Mission management subsystems usually have requirements stipulating the maximum time allowed to generate a new schedule for the next period of operations (e.g. next day). There may also be time limit requirements for how long it takes to generate ad hoc and real time changes.

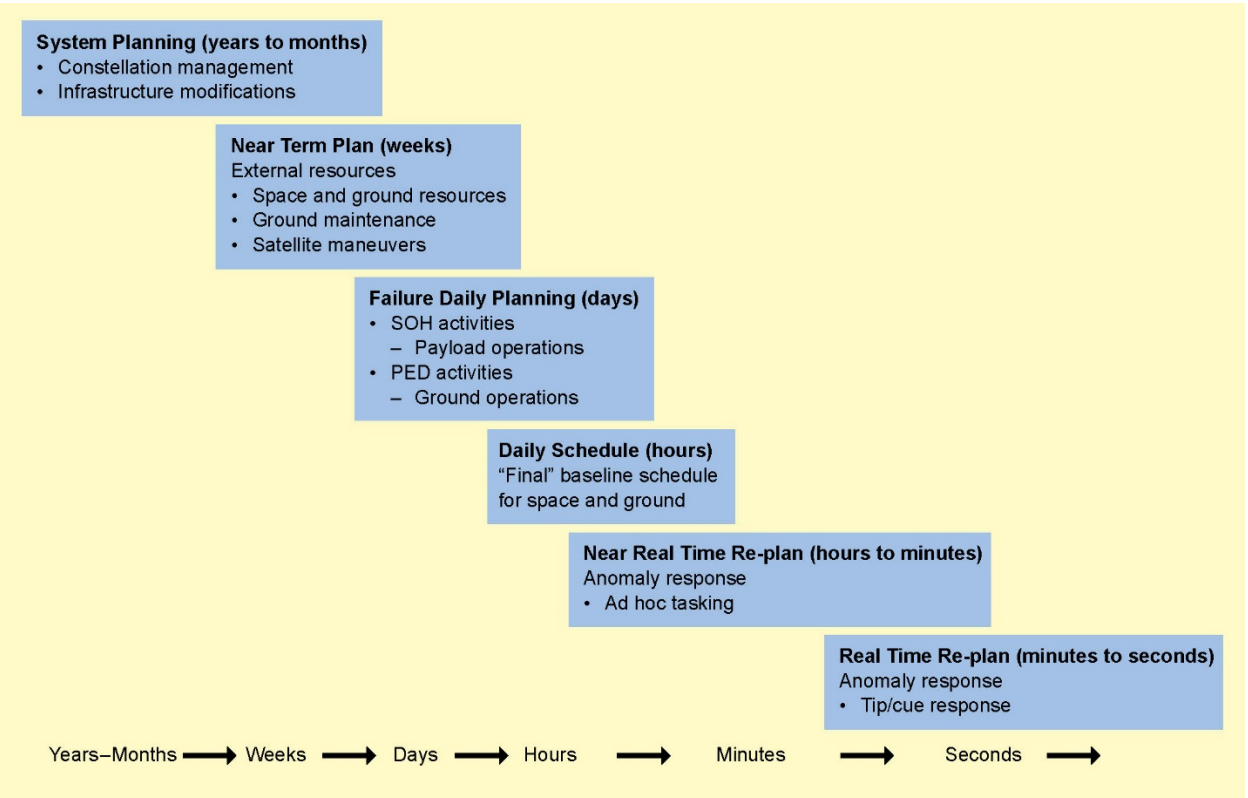


Figure 5-3. Operational mission management time frames.

5.3.3.5 Functional Flow Example

Mission management is an iterative process in which subfunctions interact. Figure 5-4 depicts the interaction of major MM subfunctions for a theoretical system that accepts tasking from the using community via a tasking infrastructure. User requests and tasking direction are accepted and stored. Requests from the tasking entity are received by the ground system's MM sub-system and turned into items to be addressed in planning and scheduling. User requests along with internal requirements, are inputs to the definition of what the system will do. This information is managed by what is called "activity management." Planning is based on the activity definitions and must take internal and external resources into account. Negotiations and/or coordination may be required with both external and internal resources to resolve conflicts or achieve better performance. Planning may also include iterations with the tasking infrastructure or users. Schedules are based on the work products from planning. Plans and schedules may be provided to the tasking infrastructure and users. Once schedule execution begins, events (including internally identified, as well as tasking infrastructure input) may require replanning, resulting in altered schedule generation and promotion. Such changes affect future planning. The executed schedule is utilized in MM and total system performance analyses and may be provided to the tasking infrastructure and users.

In Figure 5-5 the box overlay highlights the portion of the NASA OCO-2 ground system leveraged for nominal operations. The mission planning cycle includes the collection of necessary inputs to define the next set of command sequences for the flight system and its payloads to be uploaded to the spacecraft. As shown in the figure, the downlink signal is decommutated, and global positioning satellite (GPS) data, science (payload) telemetry, and engineering telemetry are extracted and processed. The payload and engineering telemetry is then ingested by the functions realtime monitoring, science data processing, spacecraft performance analysis, and instrument performance analysis, and vital instrument and spacecraft health and status are determined. Using the GPS data (satellite location) and the health and status information, a new mission plan is crafted through the set of predefined procedures and processes established during the pre-planning and commissioning phases. This includes the creation of any navigation or maneuver plans, science (payload) plans, and then the command sequences and necessary real-time commands to successfully execute uplink to the spacecraft.

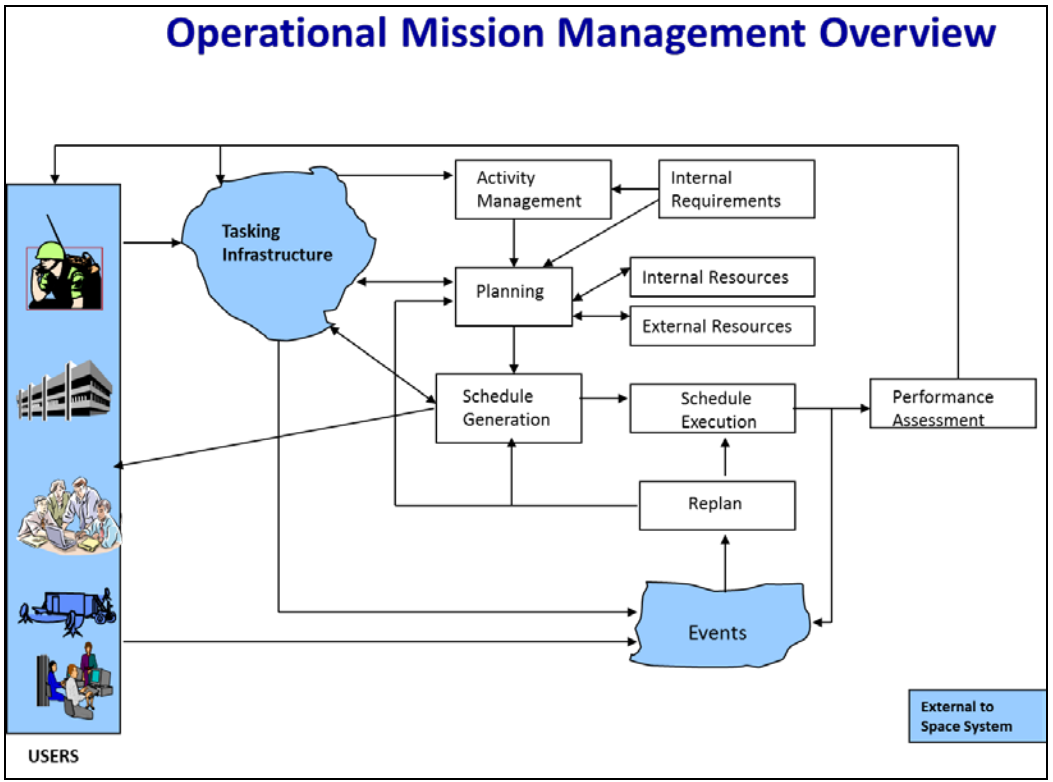


Figure 5-4. Operational mission management overview example.

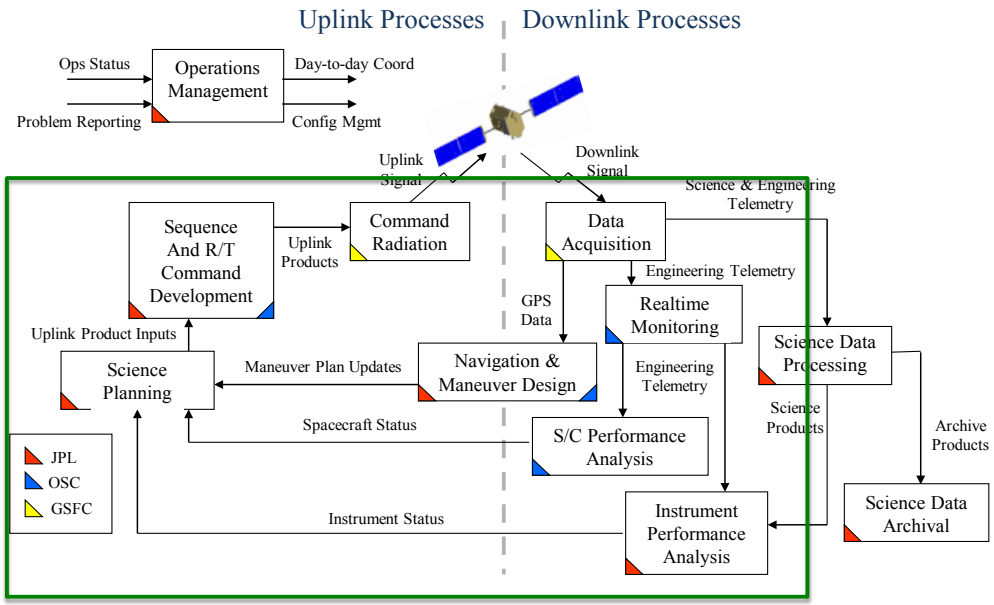


Image courtesy of NASA/JPL

Figure 5-5. Mission operations overview, NASA mission, orbiting carbon observatory – 2.

Operational MM includes a functional view of system planning; constellation management; payload planning, tasking, and scheduling; housekeeping planning and scheduling; external resource scheduling; maintenance planning and scheduling; processing and distribution planning and scheduling; ground resource planning and scheduling; schedule dissemination; tasking feedback; performance assessment/reporting; and report generation.

5.3.3.6 Operator Support

MM operators utilize a variety of displays to plan, schedule, and monitor. Scheduling operators are often day shift personnel. In such operations, schedules for weekend days are generated during the previous week prior to the weekend. Some space systems include 24/7 MM operators in their flight crews to monitor execution and respond to real-time situations; MM-provided tools are utilized to display and manage database information. These allow operators to do things such as define new tasks, modify and/or override task parameters, and group tasks for consideration. Map projections are commonly used in planning and can include orbital information as well as payload viewing and/or access capabilities. Such views can be at the constellation level as well as for a single asset. Schedule summaries are often depicted in timeline displays, which show parallel timelines for multiple satellite subsystems and payloads that include multiple satellites and ground activities.

5.3.4 Decommissioning

Decommissioning comes at the end of the usefulness of the asset. Existing MM and OD tools and subsystems may be utilized during this phase. Decommissioning can be due to "end-of-life" situations where on-board systems fail and their ability to support any form of useful mission activities is significantly degraded. However it may also be due to lack of funding to continue asset operations. United States' policy requires that satellites must be disposed of in a controlled and safe manner. This implies either boosting to a non-interfering orbit (such as super-synchronous for geosynchronous belt assets) or executing a controlled reentry into the Earth's atmosphere. Planning for such maneuvers is similar to that conducted during commissioning. Planning the shutdown of space asset subsystems is somewhat the reverse of planning the activations during commissioning, but includes exhaustion of propellant, depletion of batteries, passivation of vehicle control systems, and turning off control processors. Some systems have a procedure that enables a switch that prevents future reactivation. There is not much emphasis on scheduling during this phase, but operational concepts or protocols and coordination with external resources might necessitate more rigorous scheduling for commanding.

5.3.5 Orbit Determination

Space systems MM is heavily dependent upon OD, which is a technique for estimating the path of an object (such as a spacecraft) based on tracking measurements or observations. Future positions and/or velocities of the spacecraft are estimated. For mission planning, the OD is used for many purposes, including maneuver planning (delta V's), collision avoidance predictions, antenna pointing for crosslinks or from the ground, and sensor collection planning and processing. Estimating where the spacecraft was, is, or will be at a particular point in time is critical from planning stages, all the way through mission execution and processing of mission data during or after a sensor collect.

The OD consists of the six orbital elements or, equivalently, the six components of the position and velocity as a function of time, commonly referred to as the state vector or ephemeris. The state vector is usually defined in a Cartesian reference frame relative to the Earth's center of mass. The spacecraft's motion is approximated by a set of equations of motion (reference orbit) with the state vector adjusted in response to a set of discrete observations (or tracking) and subject to both random and systematic errors to define the true orbit, as illustrated in Figure 5-6.

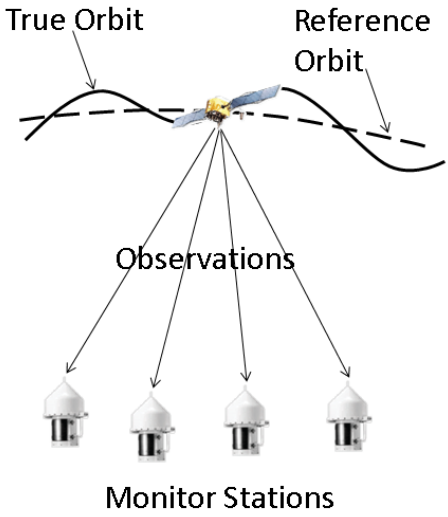


Image © The John Hopkins University Applied Physics Laboratory.

Figure 5-6. Estimating the true orbit of a spacecraft.

In addition to the planned delta-V maneuvers, the spacecraft is influenced by a variety of external forces, including gravity, atmospheric drag, third-body

perturbations, solar radiation pressure, tidal effects, and general relativity. These forces are mostly nonlinear functions of the state vector. Since the OD equations of motion are nonlinear, linear estimation techniques can be used over a short orbit arc and then combined using batch or sequential statistical estimation techniques to describe the complete orbit.

The objective of the OD is to obtain a true orbit that includes the dynamic environment in which the motion occurs with all relevant forces affecting the spacecraft's motion. Through this process, a preliminary orbit is estimated using a minimum number of observations. This estimate delivers the initial conditions that feed into the numerical integration of the nonlinear differential equations of motion that eventually produce the reference orbit. The reference orbit is then iteratively corrected using differential correction procedure and refined into the final orbit solution. An improved orbit is then calculated factoring in many observations along with an accurate physics-based model describing the dynamic environment. Once the OD is produced, mathematical propagation techniques are used to predict the future positions of the spacecraft. As time progresses, the true orbit will diverge from this predicted path (even more so with LEO spacecraft encountering significant atmospheric drag), and a new OD will need to be produced. Figure 5-7 illustrates considered inputs for OD determination and specific orbit output.

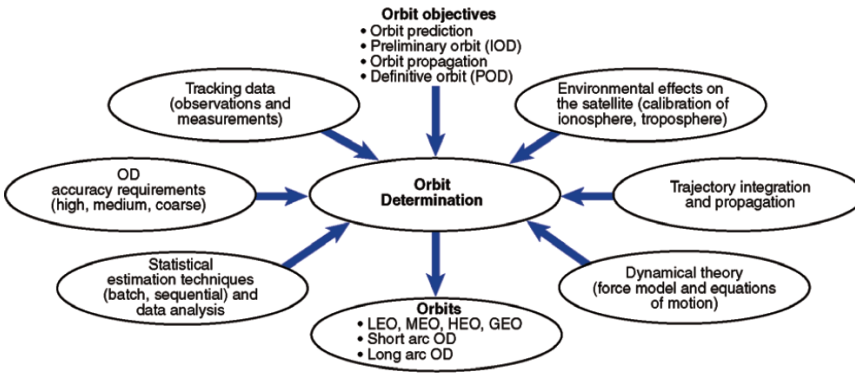


Figure 5-7. The components of the OD.

The quality of the OD will be determined by effectiveness of three main components: the models of the environment and physical laws, the observations and/or tracking data, and the computational techniques employed. For some applications, such as ground antenna pointing with azimuth and elevation, a coarse OD may be acceptable, while a more precise OD is desired for delta-V's or some sensor collection applications. Since the processing time to compute the

coarse OD will be much faster than the precise OD, developers should take that into account when deciding which type of OD to use. Figure 5-8 is an example of the OD process illustrating standard output products dependence on the accuracy of the force models.

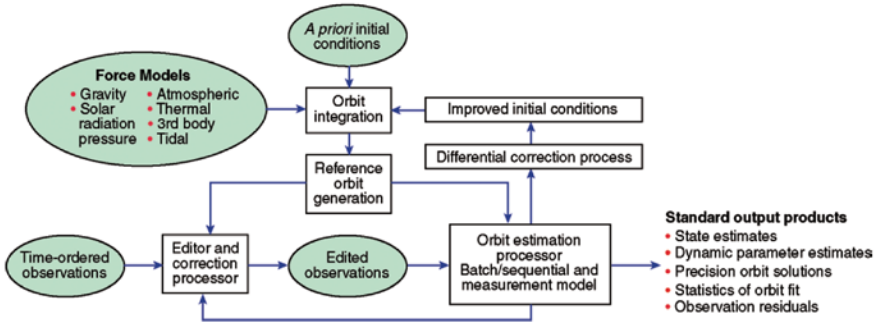


Figure 5-8. Example of the OD process.

For observational data, accuracies have improved drastically from the days of Sputnik in the 1950's with accuracies ranging from several kilometers to hundreds of meters using mostly visual observations of time-tagged azimuth and elevation. Improvements in the performance of Doppler techniques, laser ranging, and radio tracking techniques have produced current precision three dimensional accuracies ranging from two to five centimeters.

5.4 Technical Considerations

The most critical technical consideration for MM is the interaction with other major ground system subsystems, especially with space/ground asset command and control. The division of responsibilities, operational concepts, and timelines among the subsystems must be established and continually managed.

As with most space system ground systems functional areas, requirements for capacity, speed, growth and/or change pose challenges for MM. The number of assets (space and ground) to accommodate and the number of assets over which optimization takes place are important drivers of design. Time allowed to react in or near real time is often difficult to meet. The magnitude and volume of tasks to consider can pose a challenge. The choice of a suitable scheduling tool and optimization methods involve technical decisions of significant criticality which can affect system performance as a whole. A relevant question involves what level of fidelity is required in the scheduling software to model the space and ground assets with sufficient accuracy? Those acquiring and developing MM

should consider the needs of the operators. What tools will they need to accomplish their responsibilities in the required timelines?

Mission management may have to provide a means to accommodate multiple classification levels in the information associated with user requests if the tasking infrastructure does not account for that.

5.5 Programmatic Considerations

In spite of the delicate interactions between MM and space/ground command and control, these subsystems are often acquired under separate contracts with different developing organizations and sometimes differing acquisition organizations. The dependencies must be actively worked throughout development and operations. National security space (NSS) communications space systems, as they currently are operated, are a good example. The bus operations are essentially separate from payload operations. Payload tasking is primarily under the control of the various organizations and/or areas that are serviced by the communications space system. The ground software (SW) system that supports the payload tasking, and hence influences its scheduling, is potentially separately acquired and maintained from the SW that supports bus operations. This separation can be seen in other space systems as well, including civil and commercial systems. Unlike NSS, NASA generally acquires its ground system as part of a mission and/or project award process, which results in the acquisition of MM and civil and commercial capabilities managed by one NASA institution.

Tasking infrastructures often support a system of systems. Such infrastructures are a major acquisition separate from all the systems they provide a tasking mechanism for. Their milestones, upgrades, and changes are independent from any of the space systems affected. The effort necessary to stay in sync and compatible with the tasking infrastructure should be a consideration in contracting for MM subsystems.

In systems where direct input to MM from users is allowed, it is important for rules of engagement to be formally established. These protocols and the associated operational concepts for this interaction should be documented.

All the functionality included in MM does not usually reside in one MM subsystem. Most ground systems do include a MM subsystem that supports planning and scheduling for operations. Capabilities of other internal tools and external systems may provide other important functions. For example, orbit determination is usually separate from the MM subsystem.

Government-off-the-shelf (GOTS) and commercial-off-the-shelf (COTS) SW is available for MM subsystems. A framework and/or infrastructure, as well as

necessary functionality, is included in such software packages. Mission-unique tailoring and additions are normally required to develop a complete ground system MM subsystem. Examples of such GOTS and COTS packages are:

GOTS

- Virtual Mission Operations Center (VMOC), developed and distributed by the Naval Research Laboratory

COTS

- EQUINOX, developed and marketed by Raytheon Intelligence and Information Systems (IIS)
- *flexplan*, developed and marketed by GMV

For the generation of highly accurate ODs and the analysis of precise observations, commercial industry, universities and government agencies have produced a wide variety of software packages mostly for offline use. Some of these packages were designed for specific satellite constellations such as GPS, and others support a variety of satellite constellations with variable inputs for orbital characteristics such as altitude, inclination, and eccentricity. Table 5-2 provides a summary of the major OD software programs available for community use.

Table 5-2. Major Orbit Determination Software Programs

Program	Organization (sponsor)	Data types useable	Filter type	Models handled and integrator	Primary application	PC- or mainframe (MF)-based	Program capabilities
CELEST	NSWC/DL (Navy) (1965)	Doppler	Batch LS		Transit, GPS	MF	Multi-arc, multi-satellite
GEODYN II	NASA/GSPC (1984)	All data types	Batch LS	11th-order Cowell predictor-corrector	All satellite types for POD and geophysics	Cyber205 Fortran-based	Multi-arc multi-satellite
GIPSY/OASIS II Real-Time GIPSY	NASA/JPL (1990)	GPS, SLR, DORIS	SRIF/SRIS	High-order Adams predictor-corrector	High-precision orbit types with GPS receivers	UNIX WKS	Multi-arc, multi-satellite
GTDS	NASA/GSPC (1975)	All data types	Batch LS	4th-order Runga Kutta, Cowell Adams predictor-corrector	NASA operational satellites, analytic and research support	MF/Fortan, R&D (PC/WIN), VAX, Scn IBM MF, SG WKS	Multi-arc, multi-satellite (50) solve for parameters
MCS	USAF/ SPACECOM (1987)	GPS pseudo-range (PR) or carrier phase	Partitioned six-state LS filter only	High-order predictor-corrector	GPS	MF or PC	Fixed-state partition
MicroCosm	VMS, Inc. (1990)	All data types	Batch LS	Cowell predictor-corrector	All satellite types	UNIX, VAX, or PC	No multi-arc capability
OCEANS	NRL (1996)	Laser PR Carrier-phase R A E (range, azimuth, elevation)	Batch KF (GPS)	Cowell 4th-order Runga Kutta, 9th-order predictor-corrector	Covariance studies, research applications	PC	Multi-arc, multi-satellite
OIP/ODP	APL (Navy) (1960)	Doppler	Batch LS	4th-order Runga Kutta	Transit	MF	Multi-arc, single satellite
OMNIS (GPS)	NGA (National Imagery and Mapping Agency [NIMA]; Defense Mapping Agency [DMA]) (1987)	GPS PR or carrier phase	SRIF/SRIS MiniBatch	Cowell predictor-corrector	GPS or satellite vehicle (SV) with GPS receiver (GPSr)	RISC6000 and SuperMini	Multi-arc, epoch state
OMNIS (GPS)	NSWC/DL (Navy) (1987)	PR or carrier phase	SRIF/SRIS, Mini-hatch	Cowell predictor-corrector	GPS or SV with GPSr	UNIX and RISC6000	Multi-arc, epoch state
STK Version 5	Analytical Graphics (2007)	All data types	Optimal Kalman filter and fixed-interval smoother	Ringa Kutta, Gauss-Jackson	All satellite types	UNIX and PC	Multi-satellite
TRACE	Aerospace Corp. (Air Force) (1960)	R,A,E GPS PR, Doppler, range rate, optical data	SRIF/SRIS, Sequential batch LS	10th-order Gauss-Jackson w/regulated time option	General analysis of operational systems and evaluation of prototype systems	UNIX, WKS, and PC	Multi-satellite (60), 1000 estimated parameters 200 tracking stations
UTOPIA, MSODP	University of Texas at Austin, Center for Space Research (1990)	Laser, altimeter, range-rate (one- and two-way), GPS, Doppler	SRIF/SRIS	Fixed-step, fixed-order integrator	POD	Cray, HP, UNIX workstation	UTOPIA for single satellite, MSODP for multi-satellite

Special-purpose OD programs Operational OD programs for all data types GPS operational programs.

5.6 References

1. Benator, Sherrie et al. “Ground Systems for Satellite Operations: Primer and Acquisition Considerations,” Ground Systems Architecture Workshop Tutorial, March 27, 2006

2. Vetter, Jerome R.: “Fifty Years of Orbit Determination: Development of Modern Astrodynamics Methods” from *Johns Hopkins APL Technical Digest*, Vol 27, Number 3 (2007), pages 239-252.

5.7 Acronyms

AFSCN	Air Force Satellite Control Network
APL	Applied Physics Laboratory
BOL	beginning of life
COTS	commercial off-the-shelf
C&C	command and control
D/L	downlink
DMA	Defense Mapping Agency
DOD	Department of Defense
DORIS	doppler orbitography & radiopositioning integrated by satellite
GEO	geostationary earth orbit
GIPSY/OASIS	GPS-Infrared Positioning System Orbit Analysis Simulation Software
GOTS	government off-the-shelf
GPS	global positioning system
GPSr	global positioning system receiver
GS	ground station
GSFC	Goddard Space Flight Center
GTDS	Goddard Trajectory Determination System
HEO	highly elliptical orbit
HP	Hewlett-Packard
IBM	International Business Machines Corporation
IIS	intelligence and information systems
IOD	initial orbit determination
IOT	in-orbit test
JPL	Jet Propulsion Laboratory
KF	Kalman Filter
LEO	low Earth orbit
LEO	launch and early orbit
LS	least squares
MCS	master control station
MEO	medium Earth orbit
MF	main frame
MM	mission management

MSODP	multi-satellite orbit determination program
NASA	National Aeronautics and Space Administration
NGA	National Geospatial Intelligence Agency
NIMA	National Imagery and Mapping Agency
NRL	Naval Research Laboratory
NSS	National Security Space
NSWC/DL	Naval Surface Warfare Center/Dahlgren Laboratory
OCO	Orbiting Carbon Observatory
OD	orbit determination
OIP/ODP	orbit improvement program/orbit determination program
OOT	on-orbit test
OSC	Orbital Science Corporation
OTDS	orbital tracking and determination system
PC	personal computer
PED	processing, exploitation, dissemination
POD	precision orbit determination
PR	pseudo-range
R, A, E	range, azimuth, elevation
R&D	research and development
RISC	reduced instruction set computer
R/T	real time
S/C	spacecraft
SG	silicon graphics
SLR	satellite laser ranging
SOH	state of health
SRIF	square root information filter
SRIS	square root information smoother
STK	satellite tool kit
SV	satellite vehicle
SV	space vehicle
TRACE	trajectory analysis and orbit determination program
UNIX WKS	Unix workstation
USAF	United States Air Force
USN	Untied Space Network
UTOPIA	University of Texas Orbit Processor
VAX	virtual address extension
VMOC	virtual mission operations center
VMS	Van Martin Systems, Incorporated
WIN	Windows

Chapter 6
Space/Ground Asset Command and Control

Geraldine A. Chaudhri
Software Systems Assurance Department
Computer Applications and Assurance Subdivision
Timothy J. Spinney
Flight Operations Integration and Engineering
Space Innovation Directorate

6.1 Introduction/Background

The space/ground asset command and control subsystem may be considered the heart of the ground segment, as it plays a vital role in the command and control of the space vehicle and the ground assets. This chapter describes the functions of a typical space/ground asset command and control subsystem and its interfaces with other subsystems of the ground segment. It provides detailed technical data relating to specific types of commands and telemetry and how each is processed. Technical and programmatic considerations are discussed as they relate to the acquisition of a space/ground asset command and control subsystem.

6.2 Definitions

ASCII is the American Standard Code for Information Interchange.

Bus data is the health and status data that originates from the space vehicle. In some cases, bus data will include health and status data originating with the payload and included in the bus telemetry. The bus is the infrastructure of a space vehicle, usually providing locations, power, thermal, and communication for the payload.

Commercial Off-The-Shelf (COTS) is a term that references non-developmental items (NDI) sold in the commercial marketplace and used or obtained through government contracts. The set of rules for COTS is defined by the Federal Acquisition Regulation (FAR). A COTS product is usually a computer hardware or software product tailored for specific use and made available to the general public. Such products are designed to be readily available and user friendly.

Consultative Committee for Space Data Systems (CCSDS) is a multi-national forum for the development of communications and data systems standards for spaceflight.

Extensible Markup Language (XML) Telemetric and Command Exchange (XTCE) is an XML based data exchange format for space vehicle telemetry and command meta-data. Using XTCE, the format and content of a space system's command and telemetry links can be readily exchanged between space vehicle operators and manufacturers. This standard was created by the Object Management Group's (OMG) Space Domain Task Force (SDTF) and is accepted as a Blue Book standard by CCSDS. Blue book standards are standards recommended by the CCSDS that define specific interfaces technical capabilities or protocols, or prescriptive and/or normative definitions of interfaces, protocols, or other controlling standards such as encoding approaches.

Front End Processor (FEP) is a dedicated device (usually a computer) which performs preliminary processing of data prior to being sent to the main processor for further analysis or processing. The FEP performs such tasks such as demodulation, bit synchronization, decoding and frame synchronization for the telemetry stream and modulation and encoding for the command stream.

Health and Status Data includes configuration status and housekeeping data from various space vehicle subsystem units and operational instruments.

Mission Data are produced by the payload instrumentation and communicated to the ground.

Object Management Group (OMG) is an international, open membership, not-for-profit technology standards consortium. Founded in 1989, OMG standards are driven by vendors, end-users, academic institutions, and government agencies. The group within OMG most involved with space telemetry and command is the Space Domain Task Force (SDTF), which has created standards for use in space systems.

Time Division Multiplexed (TDM) is a method of transmitting and receiving independent signals over a common signal path by means of synchronized switches at each end of the transmission line so that each signal appears on the line only a fraction of time in an alternating pattern. This strategy is most often implemented as a fixed set of messages, organized into a fixed sequence, which is continuously repeated.

6.3 Space/Ground Asset Command and Control Subsystem Overview

The main function of the space/ground asset command and control subsystem is to format the commands for transmission to the space vehicle and payload and to process the telemetry received from both the space vehicle and the payload. Another important function of this subsystem is to monitor and control the

ground equipment. In some cases, the space/ground asset command and control subsystem of the payload may be handled by a separate ground segment. Telemetry data includes both bus, payload health and safety data and mission data. Bus data is the health and status data that originates from the space vehicle that may include payload health and safety data. Mission data is strictly data from the payload sensors and other payload instruments. Figure 6-1 highlights the space/ground asset command and control subsystem within the ground segment reference architecture.

Efficient ground segment architecture begins with the communication backbone. Current ground segment implementations typically use a network to provide communication between the subsystems. If the ground segment supports multiple levels of data classification (e.g. classified mission data delivered as an encrypted stream over the unclassified Air Force Satellite Control Network [AFSCN]), multiple communication backbones may be necessary to support services in the different classification enclaves. When multiple security enclaves are in use, cross-domain solutions are used to assure the appropriate transfer of data between the enclaves.

The space/ground asset command and control subsystem interacts with the other subsystems of the ground segment over the network, using infrastructure services. Modern implementations may have actual services, while older implementations may mimic services using some combination of function invocations and predefined connections.

For telemetry, data flows from the space vehicle through the ground terminal to the space/ground asset command and control subsystem. Raw data is distributed to the archival and retrieval function for future use. The raw data is decommutated and converted to engineering units, a process also known as calibration. This processed data is also distributed to the archival and retrieval function. A subset of this data is sent to the data analysis function. At a minimum, the data analysis function creates an orbit estimate for the vehicle and passes that information to both the space/ground asset command and control subsystem and the mission planning subsystem.

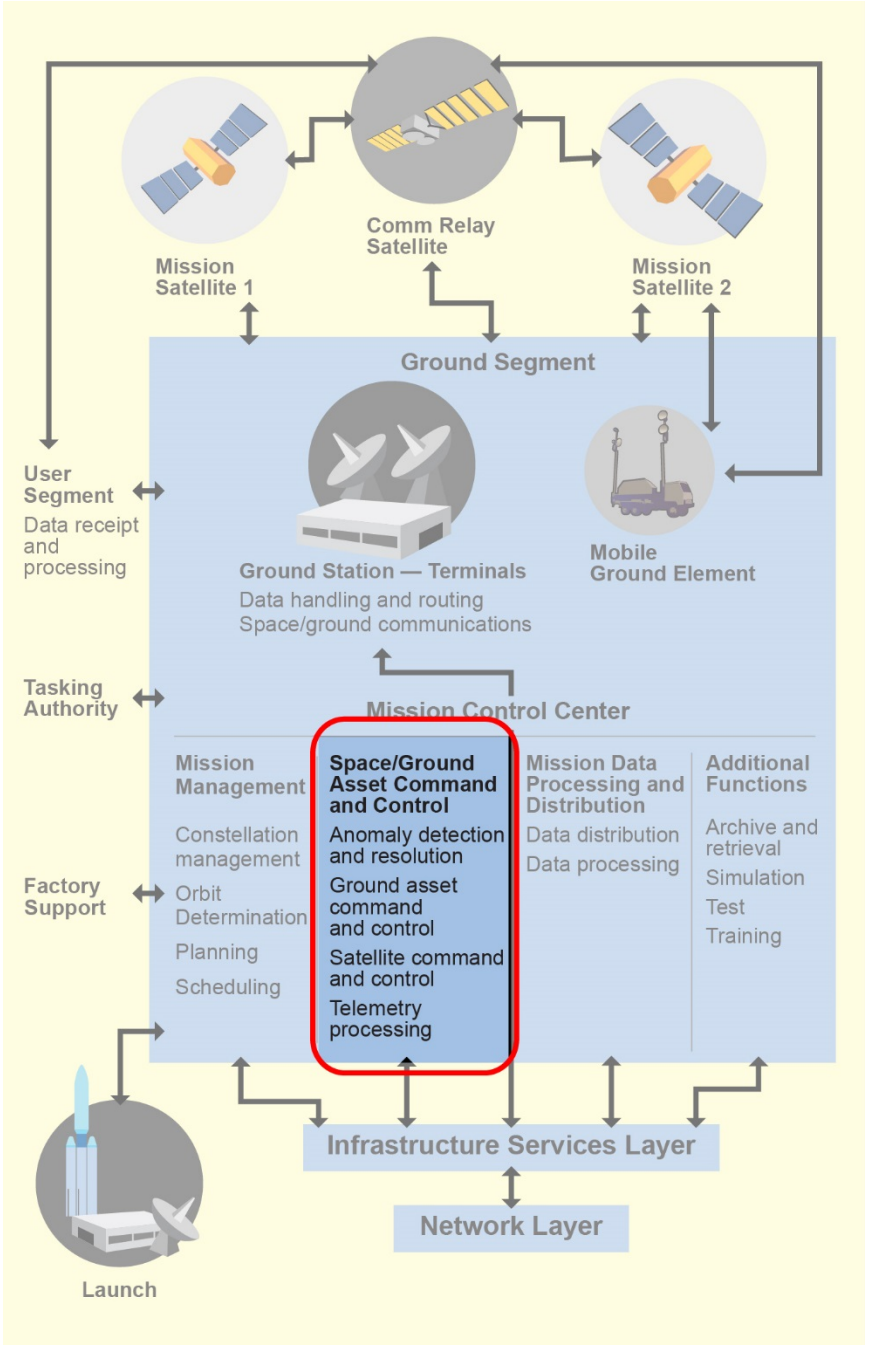


Figure 6-1. Ground segment reference architecture.

Mission management takes vehicle state and orbit estimation and determines all of the relevant visibilities and environmental conditions for the space vehicle. This information is fed to the planning function of mission management to create operational schedules for the vehicles. These schedules and the space vehicle state/environment are used to create the command sequences to be loaded to the vehicle. Although it is called mission planning, this type of planning mainly applies to the space vehicle. For space vehicles that have communication, sensor or information-gathering payloads, the mission management subsystem includes planning and scheduling to handle the management of these resources.

The schedule and command sequences are delivered to the space/ground asset command and control subsystem. Using the estimated orbit, delivered schedule, and the command sequences, the space/ground asset command and control subsystem configures the ground terminals for the contact and loads the command sequences for transmission.

6.3.1 Major Functions

The major functions of the space/ground asset command and control subsystem are listed below:

- Procedure development
- Procedure execution
- Command formatting and transmission
- Command execution verification
- Telemetry decommutation
- Telemetry processing and analysis
- Telemetry displays and trending
- Alarm/event processing
- Anomaly detection and resolution
- Ground equipment monitor and control

The functional flow diagram presented in Figure 6-2 shows how these functions (shaded boxes) relate to each other and how data flows through the subsystem. In addition, this diagram shows the interfaces with the archive and retrieval, orbit determination, mission planning, and the simulator functions (boxes with dashed outline). The telemetry data capture function and offline data analysis are represented as solid outline boxes. The telemetry data is received from the front end processors and stored in raw form to the data archive. At the same time the telemetry is forwarded to the telemetry decommutation function. The telemetry is then processed into engineering data and analyzed for alarms and space vehicle or payload anomalies and selectively displayed to the operator. Telemetry data relating to the time and position of the space vehicle is

transmitted to the orbit determination function for analysis. The ground equipment is also monitored and its data is analyzed for alarms and anomalies. On the command side, mission plans are generated by the mission management subsystem and used as input into command procedures by the procedure development function. This development usually occurs off-line. The command procedures are executed in real time by the procedure execution function and formatted into a byte stream and transmitted to the space vehicle. Command execution verification takes place after commands are transmitted to the space vehicle. In addition, configuration commands to the ground equipment are processed and routed to the appropriate ground equipment item. Command procedures are also verified through the use of simulators as shown in the functional flow diagram. Commands and directives may also be issued at the keyboard although this function is not depicted in Figure 6-2.

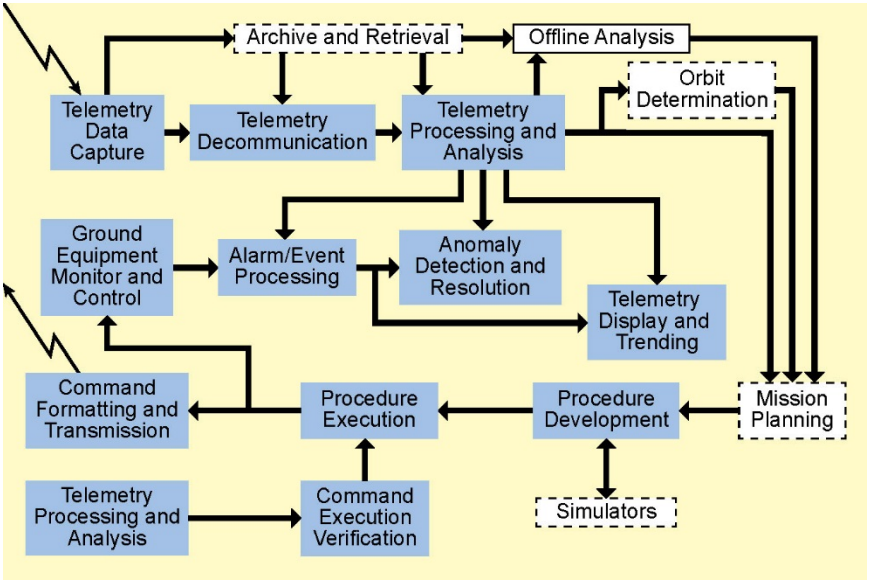


Figure 6-2. Functional flow diagram of space/ground asset command and control subsystem.

6.4 Detailed Descriptions of Functions

This section describes in detail each of the functions that make up the space/ground asset command and control subsystem.2840+1500+70+45+1000+1300+75+532+1000+750+1000+1000+2600+2000+2700+400+

6.4.1 Procedure Development

Procedure development is the capability to create scripts or files of commands that automate the activities that must be performed to command and control the space vehicle. These procedures contain sequences of instructions to conduct space vehicle operations, typically consisting of space vehicle commands and space vehicle telemetry comparisons. The procedures may also include the configuration of ground equipment, configuration of space vehicle test equipment, execution of ground testing, and execution of on-orbit testing.

A procedure or script is usually developed in an English-like language with a word processor and is executed much like a computer program. The procedure language usually contains control structures such as: "if then else," "while," "do while," "for loop," and "case statements."

There are a variety of procedure languages in commercial use today and some have actually been proposed as a standard. To develop a common standard, a minimum set of requirements must be agreed upon. This set would include the following:

- Availability of control structures (e.g., if then else, while, do while, for loop, case statements, etc.)
- Syntax checking
- Interface to databases
- User-friendly man/machine interface
- The procedure execution software must provide the following capabilities:
- Ability to set breakpoints
- Step by step execution
- Automatic execution
- User-friendly man machine interface

Some of the more common commercial procedure languages are listed below:

- Spacecraft test and operations language (STOL)
- Tool command language/tool kit (TCL/TK)
- Procedure language for users in test and operations (PLUTO)
- Comet command language (CCL)
- C-extended command interface language (CECIL) scripting language
- Spacecraft command language (SCL)
- Command interface language (CIL)
- Satellite procedure execution language and library (SPELL)

Please note that this information is provided for reference only and does not represent The Aerospace Corporation's endorsement of the products.

OMG's SDTF has launched an initiative to standardize space vehicle procedure languages with its release in 2012 of the Satellite Operations Language Meta-model (SOLM). This meta-model allows for the definition of a platform-independent model (PIM) of a space vehicle procedure. The PIM is then mapped into a platform-specific model (PSM) for procedure execution. There are numerous benefits to standardizing space vehicle procedure languages: it provides seamless transition to operations from the factory; greater ground segment interoperability, and decreases the effort to upgrade the ground segment. An added benefit of using a standard procedure language would be that ground segments could back up other segments in the case of a ground segment failure or loss of space vehicle visibility.

6.4.2 Procedure Execution

Procedure execution is the function that allows the space vehicle procedures to be processed into a command byte stream for operational control of the space vehicle. In addition, procedures are executed for configuration of ground equipment, configuration of space vehicle test equipment, execution of ground testing, or execution of on-orbit testing. There are numerous checks that take place before commands are forwarded to the front end processor. These checks are described in the section below: Command formatting and transmission. There may be varying levels of automation built into the procedure execution function. These different levels of automation are described below:

- Manual—Each line is ok'd by the operator to be executed
- Semi-Manual—Each step is ok'd by the operator to be executed
- Automatic—All steps of the procedure are executed
- Invisible—All steps are executed without any user interaction

The procedure execution function may be run on several workstations, with one workstation acting as the master controller and the others in a slave or monitoring mode.

6.4.3 Command Formatting and Transmission

Ground segments typically implement three classes of commanding: configuring the ground system equipment, sending directives to the antenna service, and commanding the vehicle. Each of these command classes uses different communication paths and possibly different communication protocols. The communication paths are illustrated in Figure 6-3.

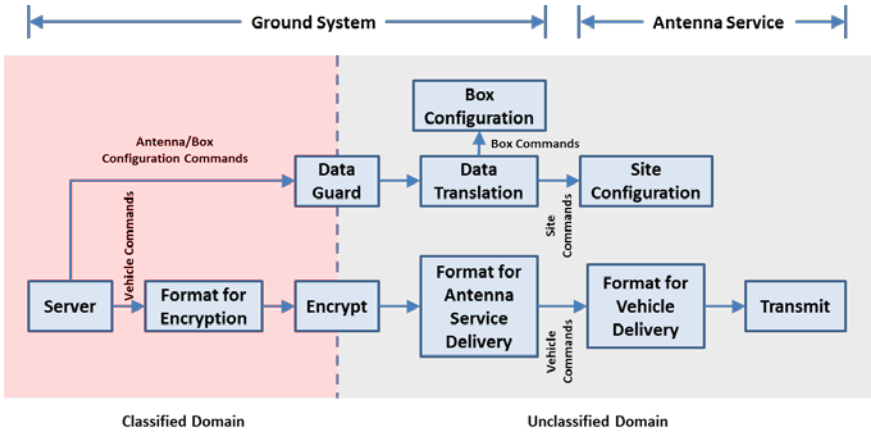


Figure 6-3. Paths for equipment/site/vehicle commanding.

Figure 6-3 shows a scenario where the ground system and the antenna service are in different classification enclaves. This is relatively common and requires additional effort to avoid contaminating the unclassified enclave. Vehicle commanding, as illustrated by the bottom row in the figure, passes from the ground system to the antenna service as an encrypted binary stream. The encryption device acts as an approved red/black separation unit.

Commanding for the antenna service and the supporting ground equipment is, in general, unclassified. However, this data is originating in a classified environment. Status information originates in an unclassified enclave, but must be passed into a classified enclave. The approved solution is to pass this commanding and status information through a data guard. The data guard assures that commands and data passing from one enclave to another meet the rules implementing the system security policy.

In many cases, antenna configuration messages are sent as XML messages to simplify passing data through the data guard. Unless the antenna service and/or the equipment in the same enclave handle the XML messages natively, the messages must be converted into a format that can be used by the respective devices. This is done in the data translation function.

“Box Configuration” involves the configuration of all of the equipment in the same classification enclave as the antenna service. It is this equipment that performs the “Format for Antenna Service Delivery” function. “Site Configuration” involves the configuration of the antenna service equipment for the “Format for Vehicle Delivery” functions.

In most situations, commanding implies verification of the command result. Verifications may be explicitly requested, or provided on a periodic basis, or derived from log file messaging. These verifications are used to drive operational responses. These responses may be manual or automated. A classic example is when a vehicle command is rejected by the vehicle. The standard response to a rejection (a negative verification) is to re-transmit the command.

Commands are built and sent from an established information store. The ground system team builds the command database for the equipment. The command (or directive—depending on system) database for the antenna service is provided by the antenna service provider. Vehicle commands are delivered in a vehicle command database from the space vehicle (and sometimes the payload) contractor. These information stores may be integrated or separate.

The processing of command data involves converting a command mnemonic and its arguments into a byte stream and then formatting the data for transmission. There are two additional important functions in the command process. The first is the pre-telemetry verification (PTV)—this ensures that the values of a list of telemetry parameters must be within a specified range before the command can be sent. This is a safety measure that can be overridden if necessary. The second is a check that values sent as command arguments are within the proper range.

All the information required by the space/ground asset command and control subsystem is stored in a database, which is delivered by the space vehicle manufacturer to the flight operations team and the ground segment developer with proper documentation. This is the source of a potential conflict that should not be underestimated. The space vehicle manufacturer uses the database with its own system for ground testing. The flight operations system might use a different space/ground asset command and control subsystem with its own database structure. The decision to convert the database to the format already in use at the control center must be weighed carefully against the risk and cost of conversion or adaptation. Using the space/ground asset command and control subsystem of the space vehicle manufacturer implies that the operations team will require additional training so that it acquires the required proficiency. Tools to manage the database (editing, version control, and so on) are also required. An effort has been made by OMG's SDTF to specify an information model for telemetry data and commands in order to standardize the exchange of databases. The resulting model, XTCE, has become a CCSDS standard and may help in improving the situation.

The process for command generation varies from organization to organization. Some organizations pre-build commands (prior to being available for execution) and then store the binary execution strings (with appropriate labeling) for transmission as required by operations. Some organizations pre-build commands

as an American Standard code for Information Interchange (ASCII) string, including the command parameter values, store the string, and convert the string to the appropriate binary string on transmission. Most systems permit the ability to build a command from the console and then execute it.

Figure 6-4 shows the process for delivering a command to the vehicle. The process starts when the command is retrieved from the command database, command arguments are checked to ensure that they are within range. Telemetry may also be checked to ensure that the space vehicle is in the proper configuration to receive the command. The command and its arguments are then converted to a binary string. This conversion is performed in accordance with the vehicle command database.

The binary command string is then routed to the data handling and routing function to be formatted for the encryption process. Formatting begins by converting the binary command string into a series of ‘n’-bit blocks (where ‘n’ is typically 64). Each of these ‘n’ bit blocks is prepended to a ‘n’-bit authentication block. Each of the resulting blocks is forwarded to the encryption process. The encrypted blocks from the encryption process are passed to a process that formats the blocks for delivery to the space/ground communications function.

The space/ground communications function determines the required format for delivery of the command blocks. Format requirements reflect the behaviors supported by the space/ground communication link. For example, if the space/ground communications function provides ternary commanding (1,0,S), the transport protocol (from the ground system to the antenna) may require dibit encoding of the command blocks (to allow embedding S-bits in the stream). Older systems may require sequence numbers for each of the command blocks (allowing for missing and out-of-order command blocks). Because all of the data to be sent to the antenna service has been encrypted, in many cases this work can be executed in an unclassified enclave.

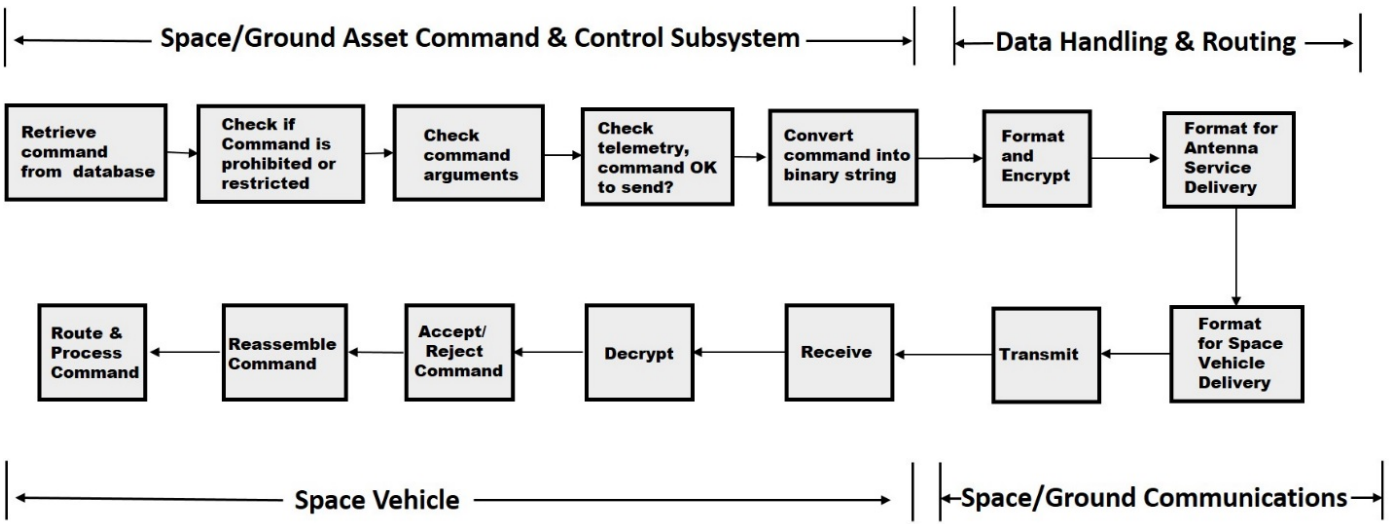


Figure 6-4. Sample command process.

When the encrypted, binary command blocks arrive at the space/ground communications function, they are converted from the transport protocol to the transmission protocol. The transmission protocol varies from mission to mission. This format conversion process can be illustrated by the following example: The incoming command blocks are buffered to resolve missing and out-of-order command blocks. Once resolved, the individual command blocks are removed from the transport protocol and converted to a ternary stream. The zero and one bits are run through a Viterbi encoder. The S-bits continue to frame the (now Viterbi encoded) command block bits. The ternary stream is then modulated onto the carrier and transmitted to the vehicle.

The space vehicle receives the modulated signal and extracts a ternary stream. This ternary stream is sent to the space vehicle decryption unit which both decrypts it and converts it to a binary stream. The authentication block for each command block is verified. If the verification fails, the command block is rejected and the reject counter is incremented. If the verification succeeds the space vehicle command count is incremented and the command block is passed to the flight software. Flight software then reassembles the commands and routes them appropriately.

Space vehicle commanding can be done in either binary or ternary commanding. Binary commanding, using the Air Force satellite control network (AFSCN) can execute at up to 100 kilobits per second (kbps). With an unusual configuration, it is possible to uplink at 256 kbps, but this requires using the main beam—losing telemetry. Because encryption requires an S-bit between command steps to perform the encryption, and binary commanding does not provide the ability to send an S-bit, users have developed the use of a Barker word to signal the S-bit.

There are two formats that are typically used for commanding the space vehicle: the communications operation procedure-1 (COP-1) format as recommended by CSSDS or Time Division Multiplexed (TDM) format. COP-1 command protocol is focused on autonomously ensuring that commands are received by the space vehicle in the order that they were transmitted and that all commands get to the space vehicle successfully. TDM command formatting has been in use since the early days of space missions. Many Department of Defense (DOD) missions still use this formatting scheme. In this scheme, data is continuously multiplexed in defined, fixed-length frames. Each mission defines their own multiplexing scheme as well as their own method for verifying that commands reach their destination.

All DOD satellites are required to have encryption on the uplink. As part of the command processing, each command step is formatted in accordance with the cryptographic device. As part of the formatting the command for encryption, the command is decomposed into a sequence of command steps. Encryption of the

commands can be in Authenticate mode or in Data mode. Authenticate mode permits the vehicle to validate each command step. This provides additional security, but at a significant cost in effective data rate. Alternatively, the ground system can encrypt using Data mode. Current best practices call for switching into and out of Data mode for each command. This switching requires the transmission of a command step (16 bytes) to enter Data mode and the transmission of a second command to exit Data mode. Authenticate and Data modes require the same amount of time to transmit a command that would fit into two Data mode command steps. If the command is larger than 32 bytes, Data mode is more bandwidth efficient than Authenticate mode. Data mode is always more efficient for large data block uploads.

When commands are sent in Authenticate mode, each command step is composed of a cryptographic data block and a command block.

Command decompositions for both Authenticate and Data Mode are presented in Figures 6-5 and 6-6. An example of a single binary string that is too large for the allowable encryption block size is shown in both Figures 6-5 and 6-6.

Figure 6-5 shows how the command string is decomposed to conform to the Authenticate mode. In this mode, half of the allowable encryption block size is allocated to the command string fragment. The other half of the allowable encryption block size is allocated to the authentication block. The authentication block is intended to preclude acceptance of out-of-order command steps.

Figure 6-6 shows that with the deletion of the authentication block, the command string fragment can be up to twice as large as in Authenticate mode. This provides twice the effective command data rate at the cost of potentially having to deal with out-of-order command steps. It is important to note that this mode requires a command to enter and to exit.

Both figures also show the command steps framed by one or more S-bits. Without the framing S-bits the cryptographic devices have no mechanism to identify the beginning and end of any given command step. Decryption begins when the decryption input switches from an S-bit to either a zero or a one. Decryption of the command step ends on the receipt of the next S-bit.

As mentioned, there are three types of commanding possible from the space/ ground asset command and control subsystem: configuring the ground system equipment; sending directives to the antenna service; and commanding the vehicle. For most ground segments, all of this commanding is issued from a single port (serial or internet protocol [IP]) on the command and control server. Because all of the types of commanding use the same port, they can potentially

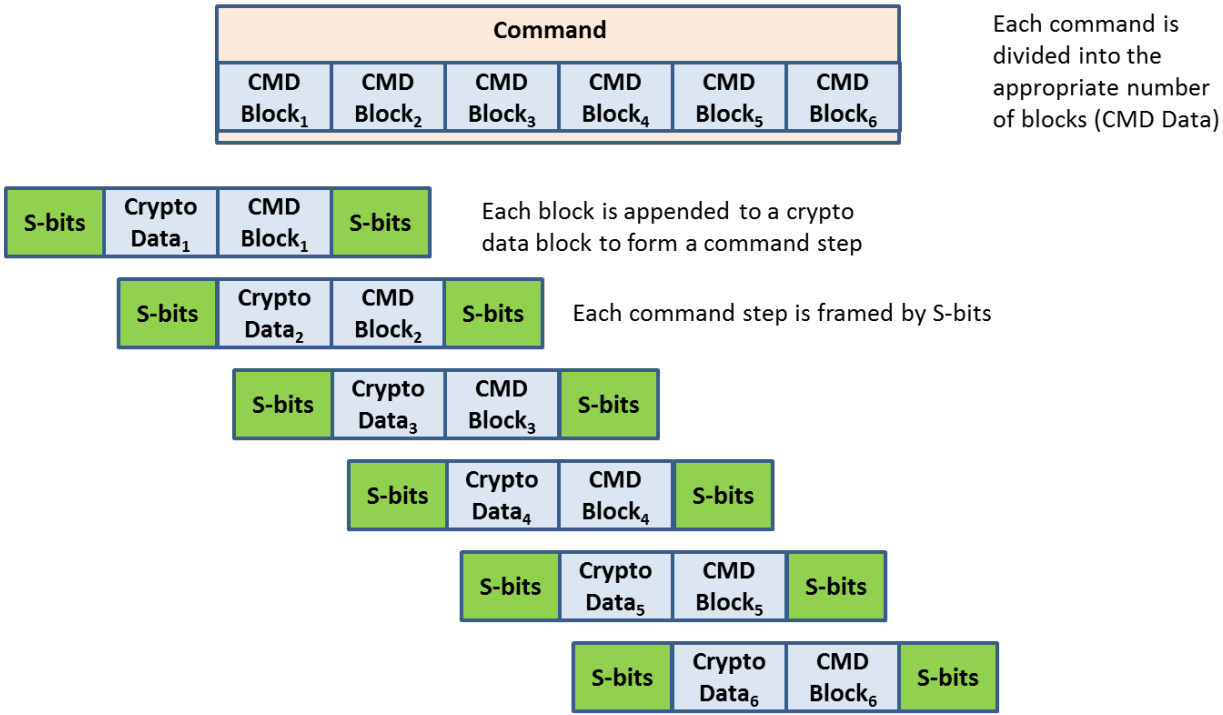


Figure 6-5. Command decomposition: authenticate mode.

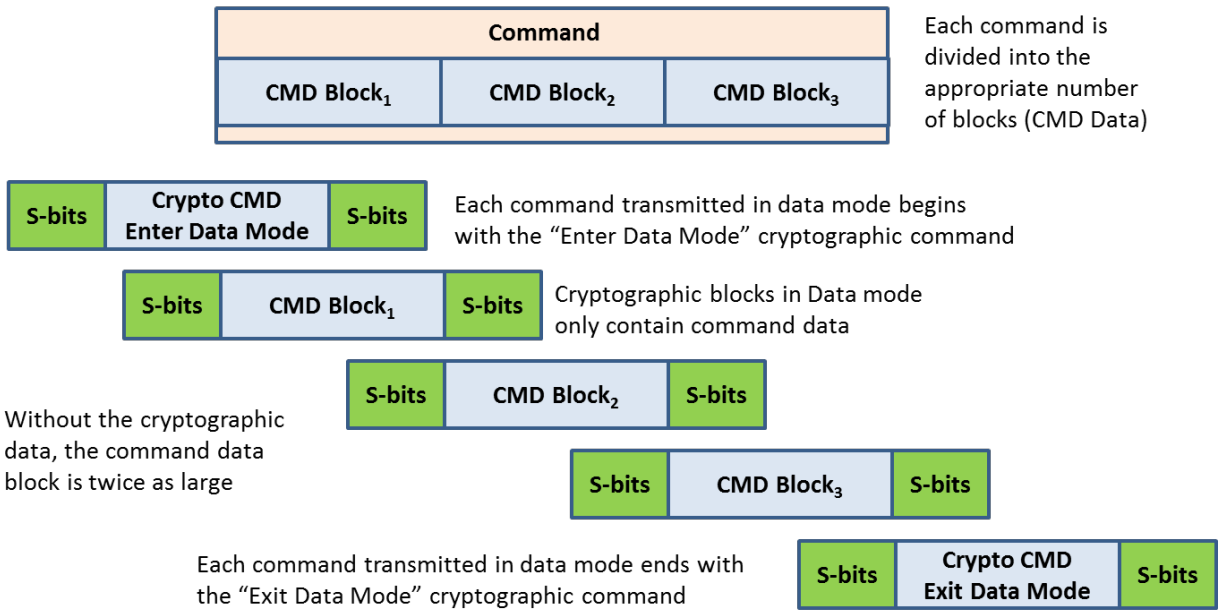


Figure 6-6. Command decomposition: data mode.

interfere with each other. This potential interference is typically dealt with by queuing the commands and executing them in either first-in-first-out or priority order. The effect is to have variable latency for the commands to a single destination.

Commands to the ground system equipment are routed over the ground system communications architecture. This architecture is generally, but not always, a local area network (LAN). Some architectural implementations provide separate paths for each of the command types. When separate paths are provided, the issue of variable latency for command release is removed.

Directives for the antenna services are routed around the encryption process and directly to the process that formats the directives for transport to the antenna services. Directives for the antenna services include antenna positioning commands, typically sent for low Earth orbit (LEO) vehicles periodically, between one and ten seconds between transmissions.

Because the antenna service does not, in general, require encrypted directives or provide encrypted status, there is a potential for providing a bypass of the vehicle encryption. To avoid this issue, an authorized data guard is used to provide the appropriate red/black separation. This same data guard is used when commanding or receiving status from the equipment on the black side of the cryptographic equipment. This architecture is briefly illustrated in Figure 6-3.

6.4.4 Command Execution Verification

The command execution verification function provides verification that the space vehicle commands were successfully transmitted and properly executed on board the space vehicle. There are several ways to verify receipt of commands by the space vehicle.

One way is through the use of the COP-1 protocol. COP-1 uses a frame operating procedure (FOP) on the command transmission end and a frame acceptance and reporting mechanism (FARM) on the receiving end. The FOP takes the command virtual channel frames that are ready for uplink and inserts a sequential frame sequence number (FSN) in the frame header prior to transmission. After transmission of the command frame to the space vehicle the on-board FARM checks to ensure that the FSN of arriving commands is sequential and validates that the incoming command frames successfully passed through the error detection/correction software. Once successful command reception is verified, the FARM generates an updated Command Link Control Word (CLCW) for downlink. The FOP interrogates the CLCW and retransmits command frames as requested. The FOP also has a timeout function that initiates command retransmission for any command frames that are not verified in the CLCW within a specified amount of time (the timeout function needs to

be adjusted as the space vehicle get further and further from Earth to take into account light-time travel delays that will lengthen the time between command uplink and receipt of an updated CLCW).

Another method of command verification is through the use of command counters and command echo. These methods are used when TDM command formatting is used to transmit commands. For TDM command streams, the command and data handling (C&DH) subsystem on board the space vehicle implements a command counter which is incremented every time a command is received. This command counter is transmitted back to the ground in the telemetry stream. In addition, the on-board C&DH subsystem will transmit the command message received or the "command echo," which can then be checked for accuracy on the ground.

With both verification methods, command execution verification (CEV) is accomplished through end item verification. End item verification is the process of identifying that the space vehicle has received a command by verifying the action associated with the command occurred. For example, a command to turn a heater on could be end-item-verified by examining telemetry to ensure the heater circuit has been turned to the "on" state and/or that nearby thermistors are registering a rise in temperature.

6.4.5 Telemetry Decommutation

Telemetry decommutation is the function by which individual parameters or measurands are identified according to their location in the bit stream. This function may be resident in a FEP. Extraction of the raw value of each telemetry parameter from the bit stream is done with the knowledge of the parameter's first bit location and its bit length. This information is defined ahead of time and stored in a database. There may be multiple definitions for the telemetry bit stream based on the format that the on-board computer uses to downlink the data. The telemetry formats therefore have to be closely coordinated with the space vehicle manufacturer.

There are essentially two methods for downlinking telemetry data: in fixed size frames or in variable length packets.

Fixed-size frames are known as TDM frames and they adhere to a rigid format of major and minor frames. Each frame contains a frame identification and a sync pattern. Telemetry parameters may occur multiple times in the same downlink frame or packet. This is known as supercommutation. On the other hand, subcommutation is when telemetry parameters occur only periodically, not in adjacent downlink frames or packets. Figure 6-7 shows an example of fixed-size telemetry frames with 4 major frames each containing 5 minor frames or subframes of 12 words.

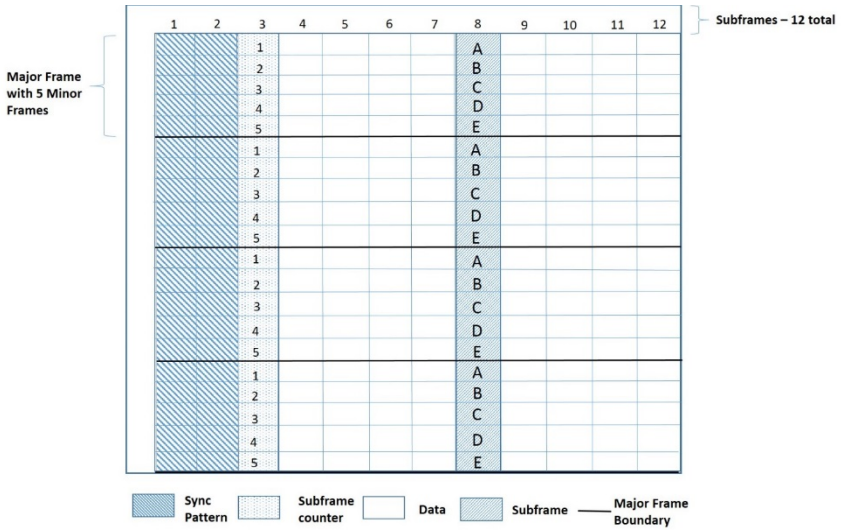


Figure 6-7. Fixed sized frames of telemetry.

Packet telemetry is variable in length and adheres to a format defined by CCSDS. The advantage of using packet telemetry is the inherent adaptability to changing data-transmission requirements. This is due to the variable-length nature of the packets and that the packet headers include fields that can be used to identify and interpret the contents of the packet. Data packets can be issued asynchronously by any data source, and the packets can be of whatever length to accommodate the data requirements at that instant. The service provided by packet telemetry is also directly compatible with higher layer application services such as networking and Internet applications. Figure 6-8 shows the format for a CCSDS telemetry packet.

Figure 6-8 shows the CCSDS packet in three primary divisions. The first of these divisions is identification of the packet (the first two octets). This information is the means by which the ground system determines the appropriate processing for the remainder of the packet. Protocol revisions are addressed with the version field. The next three fields identify the application from which the embedded data originated.

The next division (the next four octets) provides sequence control. The first two bits in this division provide the mechanism to embed objects larger than the maximum packet size. Every packet is identified as a first, continuation, or final packet. Using this flag, very large data blocks can be distributed over multiple packets. The next fourteen bits provide a mechanism to keep the arriving data blocks in order. This becomes especially important when data is originally missing and has to be retransmitted. The next two octets in this division specify

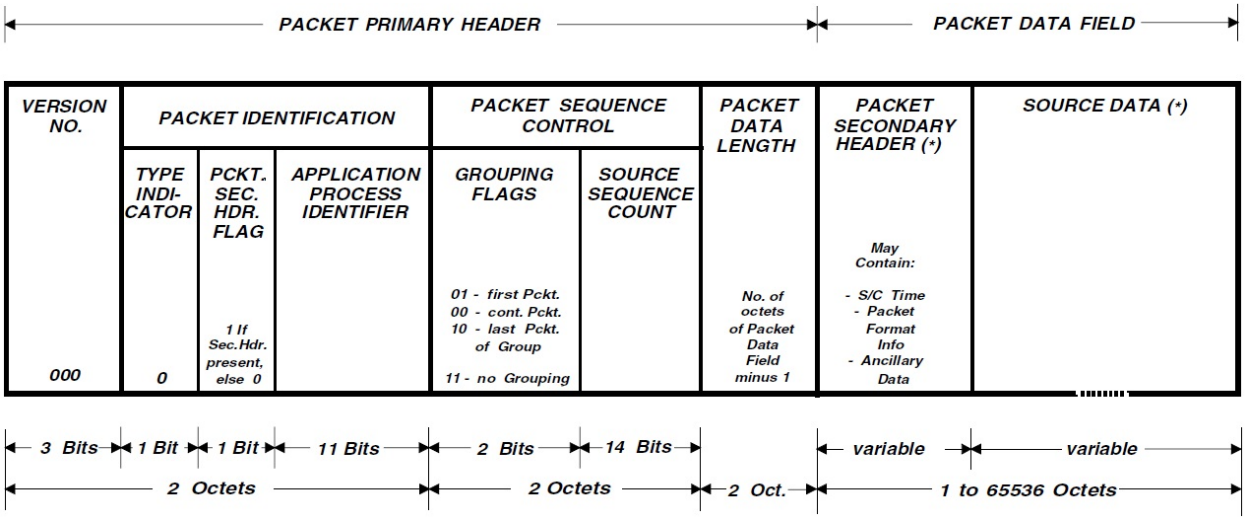


Figure 6-8. Telemetry source packet (courtesy of CCSDS).

the number of octets in the remainder of the packet. This information provides the ground system a means of determining when a packet has (putatively) been received.

The remaining octets in the packet constitute the packet data field, typically the largest division. It is consistent with the CCSDS standard for the user to define secondary packet headers and data sets within this field. Secondary packet headers can be used to simplify processing by (for example) explicitly identifying subsystems. When the packet data field is subdivided into secondary packet data fields, the user can provide separate verification mechanisms (e.g. a CRC) for each of the secondary fields. The additional verification mechanisms allow greater localization of errors (under some circumstances) resulting from transmission noise.

6.4.6 Telemetry Processing and Analysis

After the telemetry is decommutated, the following processing steps take place:

- Translation of the raw value into an engineering value—Polynomials or linear interpolation between predefined points are often used as translational aids. A discrete status is calibrated using tables (e.g., 0 = OFF, 1 = ON).
- Verification of the validity of the parameter—Parameter-specific validity conditions (as opposed to the overall validity of the bit stream verified with the cyclic redundancy check [CRC]) can be defined. For example, a reading associated with a particular item of equipment can be flagged invalid when this equipment is switched off. This is useful, for example, in situations where the last value received from the unit is down-linked although the unit itself is not delivering any more data.
- Computation of out-of-limit status—Limits are defined for all space vehicle telemetry and ground equipment data. The first limit stage, called soft alarm or warning, signals that the evolution of this value must be monitored closely and might reach potentially dangerous levels. The second limit stage, called hard alarm, signals that the value has reached a potentially dangerous level, where a corrective action must be undertaken. It is also possible to define so-called delta-limits on the difference between two consecutive values.
- Derived parameters—Parameters computed from the values of other parameters (raw or engineering). The telemetry and command subsystem usually proposes a set of mathematical functions for this purpose.

There are many mechanisms by which telemetry is organized and handled. Typically telemetry is divided into bus and mission data. The bus data represents the state of the space vehicle. This data includes configuration information (e.g.,

switch settings), recent command execution history, vehicle logs, and parameters describing the current state of some component. This bus data may also include information from the payload(s). Typical payload information captured as part of the bus telemetry includes payload voltage and current levels, temperatures, and current state.

The classes of information embedded in telemetry have very different characteristics. Some of the data is static, changing only on command. Some of the data is streaming in real-time. Some of the data has a streaming characteristic, but represents data archived on the vehicle. An example of this would be when a vehicle archives the nominal streaming telemetry during periods when it is not in contact with a ground station. Some of the data is only present as directed by user command. An example of this behavior is when the mission engineers command a subsystem to generate diagnostic data.

Many vehicles permit the mission engineers to specify the telemetry to be downloaded. This may take the form of organizing the sequence of different classes of telemetry or creating new telemetry objects.

A further complication arises with the telemetry architecture on the vehicle. Data may be stored as pre-packaged telemetry objects or as raw data to be packaged into telemetry objects. Both techniques have advantages and disadvantages. Static definition of the telemetry objects makes recovery of a specific telemetry object considerably easier for the ground system. However, static definition of the telemetry objections increases the storage overhead for the vehicle.

Telemetry data may be collected in buffers or in files. Buffers tend to work best for streaming data flows (continuously updating sets of parameter values). Files tend to work best for discrete data collections (diagnostic data from a subsystem). Several classes of bus data can, with little difficulty, fall into either architecture (log files or command histories). In many cases, the vehicle telemetry architecture is selected based on the operational concept for the vehicle. For example, geosynchronous vehicles tend to be streaming data. This is driven by a relatively low data rate and by continuous visibility. These same factors tend to make recovery of individual lost telemetry objects less important.

For LEO vehicles, contact times are shorter and less frequent. However, the data rate tends to be higher. This combination tends to lead to more file-like data transfers. This combination also tends to make the recovery of individual telemetry objects more important (since significant thinning has generally already occurred).

Notions of sequence versus time

Satellites present telemetry in a time order (or time analog such as vehicle time code word). There is a strong temptation for ground operations to assume that this time order is inviolable and perfect. However, many factors interact to falsify this assumption. An extreme example is a vehicle reset. After a reset, the vehicle time (or its analog) returns to some default value. The data associated with the default (and following) times is later than previous data (in spite of the earlier times in the parameter stream).

The ground system needs to preserve the creation order of the data. Creation order has different meanings with different data architectures (streaming versus block structured, files versus buffers). This task is significantly complicated by the inability to trust the time (or equivalent) parameter values as accurate. The process is further complicated by out-of-order downloads. These out-of-order downloads can occur because of a need for retransmission, a reprioritization of the channel bandwidth or any number of other reasons.

From an archive perspective, all downloaded parameters associated with a single download protocol packet should be grouped. Each download protocol packet must also be linked to the protocol packet immediately preceding and the packet immediately following. These linkages sometimes have to be adjusted to compensate for recovery of lost data, out-of-order data transmission, or other factors. Enforcing the concept of sequence, rather than time, on archival order permits a higher fidelity reconstruction of the data creation order and the continuously evolving state of the space vehicle memory.

6.4.7 Telemetry Displays and Trending

Once the telemetry data are processed, a display system presents the data to the flight operations engineers. The basic types of display pages are alphanumeric, where parameters are listed together with their description and value, and graphical, where the value of a parameter is plotted against time or the value of another parameter. In addition to these types of displays, alarm displays summarize the parameters with out-of-limit conditions and synoptics give a graphical overview of a subsystem. Three-dimensional animations controlled by telemetry parameters can also be used. Several sample telemetry displays are pictured in Figures 6-9 and 6-10. Figure 6-9 shows a combination of alphanumeric and graphical displays for three telemetry streams while Figure 6-10 shows a graphical display of one space vehicle's power subsystem.

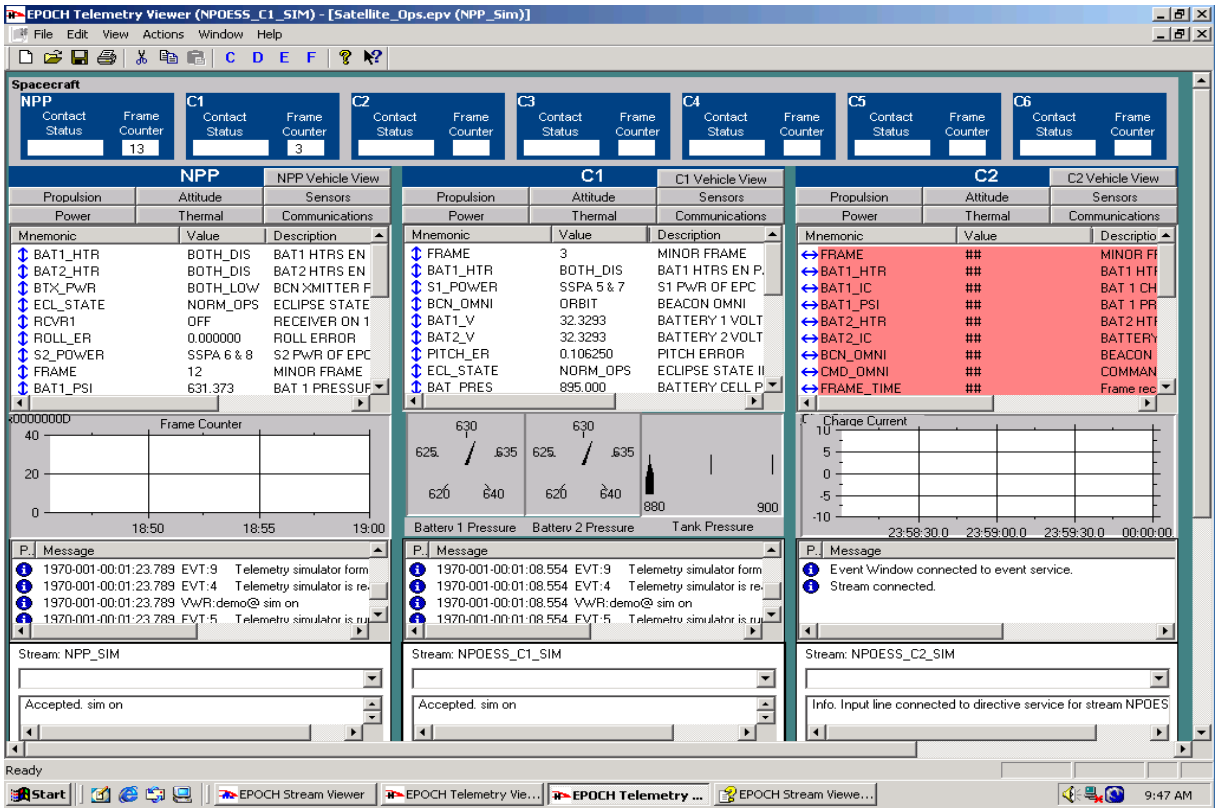


Figure 6-9. Telemetry display (Courtesy of EPOCH 2000 - Kratos).

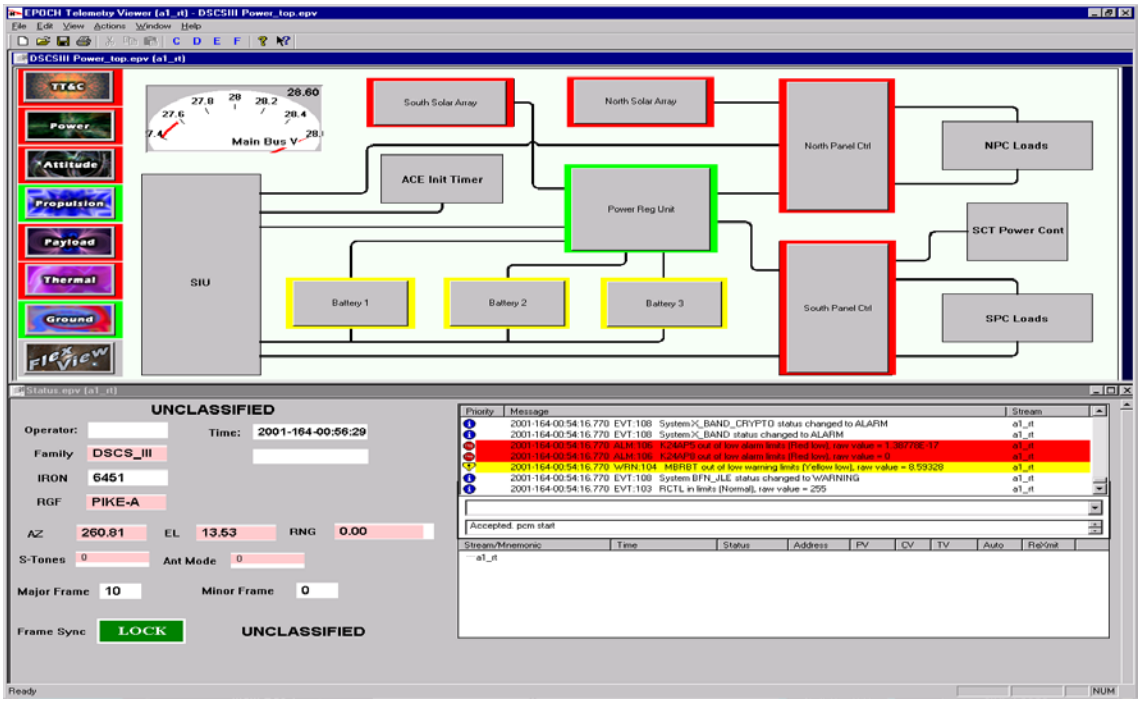


Figure 6-10. Graphical telemetry display (Courtesy of EPOCH 2000 - Kratos).

Most data trending can be accomplished using sequence, without requiring time (except as a reference for when the data was collected). Avoiding the use of time in trending significantly simplifies trending through anomaly conditions. Generally, time is important to trending where data collection times vary (are aperiodic). Generating high quality time estimates for each of the sequenced data sets can be simple when time is monotonically increasing and well correlated with a wall clock or complex after a vehicle reset with an unknown return to service time. From a database perspective, storing the telemetry parameter values (including the putative time or equivalent value) in sequence order and using the sequence index value (rather than time) as the retrieval key is preferred.

6.4.8 Alarm/Event Processing

The alarm/event processing function is responsible for monitoring the status of the space, link, and ground segments, and indicates alarms, warnings, and other events of interest. Alarms are processed based on values in the telemetry database. Alarm database limits and values may include the following: high value, low value, inclusive range, exclusive range, mask, state, change, delta, digital, and combination alarm types. These limits and values are associated with an alarm band (red, yellow, or green) in the database. When telemetry is processed in real time, the alarm database limits and values are compared against the real time telemetry values and if a red or yellow limit is reached, an alarm will be displayed. In some cases, the alarms may be audible. Figure 6-11 shows a sample telemetry alarm display.

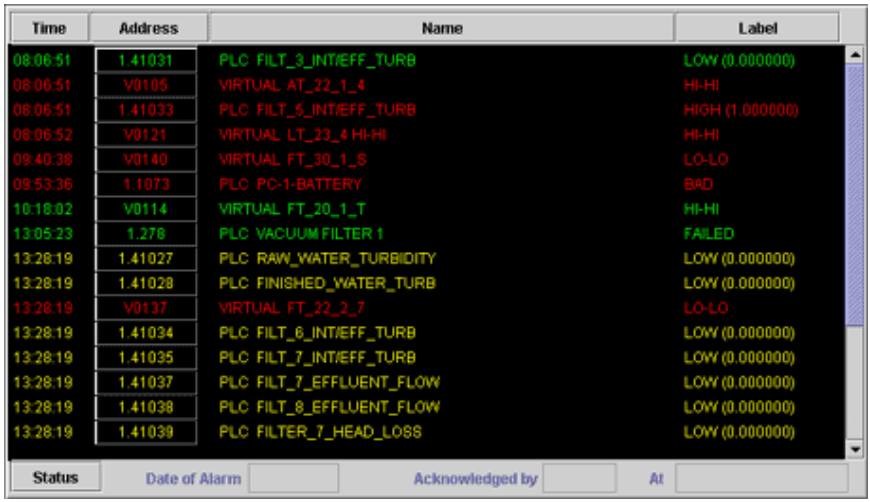


Figure 6-11. Sample telemetry alarm display (courtesy of Fr-Sky).

6.4.9 Anomaly Detection and Resolution

For anomaly detection and resolution, telemetry data is compared to conditions provided in the telemetry knowledge database related to anomaly detection and any commanded state changes. The data determines the current space vehicle state and provides anomaly messages as required to the operator. The following sub-functions are used for anomaly detection and resolution:

- display space vehicle health and status—This process provides visual display to the operator indicating space vehicle health and status and allows the operator to select display formats.
- detect anomalous condition—This process inputs verified telemetry parameters that are monitored to detect anomalous conditions in the space vehicle. Based on the anomaly detection knowledge database and an indicator of a command to change vehicle state, this process verifies that the space vehicle is configured to the intended state. If the space vehicle is not configured correctly, alarm messages are generated, indicating that a space vehicle fault has occurred.
- diagnose/isolate anomaly—This function inputs the anomalous condition and the space vehicle state. Based on a set of knowledge databases and/or procedures for diagnosing a detected diagnosed space vehicle fault, this sub-function provides a diagnosis for the anomaly.
- resolve anomaly—This function inputs the diagnosed condition and the space vehicle state. Based on the anomaly resolution knowledge database, which is a set of knowledge bases and/or procedures for resolving a diagnosed anomalous condition, this process outputs a corrective action message and a recommended corrective action. The corrective action may contain a resolution procedure for the anomaly. If an immediate resolution is not available for the anomaly, the appropriate space vehicle subsystem engineers will be summoned to resolve the anomaly.

6.4.10 Ground Equipment Monitor and Control

The ground equipment monitor and control function is responsible for configuring the ground equipment and monitoring its status. Ground equipment includes: the ground terminals or antennas; the front end processors and baseband equipment; network equipment (routers and network lines); computer servers; disk arrays; user workstations; printers; and all off-line storage devices. This function will send the status of each monitored ground equipment item to the alarm/event processing and anomaly detection and resolution functions. The status data will also be displayed by the telemetry display and trending function. Configuration commands are transmitted to the ground equipment in an automated fashion through the procedure execution function or may be manually entered at the keyboard.

6.5 Programmatic Considerations

There are many commercial products on the market today that provide all of the functions required for a command and control subsystem. In essence, the functions described in this chapter are very common to most space vehicle command and control subsystems. A government program office seeking to acquire a command and control subsystem should consider buying this product as COTS. Many of the COTS products are database-driven and with some customization, a space vehicle command and control subsystem can be up and running and integrated into a ground segment. Listed below are just a few of these products:

- "Eclipse" by Raytheon
- "OS Comet" by the Harris Corporation - http://govcomm.harris.com/oscomet/
- "In Control" by L3 Comm - http://www2.l-3com.com/tw/incontrol/
- "Ace" Premier Command and Control products by Braxton - http://www.braxtontech.com/Products/command-control.html
- "EPOCH IPS" by Kratos Integral Systems International Products - http://www.integ.com/Products.html
- Neptune ™ Common Ground Architecture, ™ Navy Research Lab

Please note that this information is provided for reference only and does not represent The Aerospace Corporation's endorsement of the products.

6.6 References

1. Fortescue, P., Swinerd, G., Stark, J., *Space Systems Engineering*, Wiley. Hoboken, NV. September 19, 2011. 4th edition.

2. Wertz,J. (Editor) *Space Mission Engineering: The New SMAD (Space Technology Library*, Vol. 28) Publisher: Microcosm Press; First edition (July 29, 2011)

3. Consultative Committee for Space Data Systems (2007) CCSDS 660.0.B-1, *XML Telemetric and Command Exchange (XTCE)*, CCSDS Secretariat

4. Consultative Committee for Space Data Systems (2003) CCSDS 232.1.B-1, *Communications Operation Procedure-1*, CCSDS Secretariat

5. Consultative Committee for Space Data Systems (2003) CCSDS 133.0.B-1, Space Packet Protocol, CCSDS Secretariat

6. Sullivan, Thomas J., Alex Martinello. *Compatible Satellite C2 Prototype Common Service Concepts*. TOR-2012(1570)-11336e. The Aerospace Corporation, El Segundo, CA. September 2012.

6.7 Acronyms

AFSCN	Air Force Satellite Control Network
ASCII	American standard code for information interchange
CCL	comet command language
CCSDS	Consultative Committee for Space Data Systems
CD&H	command and data handling
CECIL	c-extended command interface language
CEV	command execution verification
CIL	command interface language
CLCN	command link control word
CMD	command
COTS	commercial off-the-shelf
CRC	cyclic redundancy check
DOD	Department of Defense
FARM	frame acceptance and reporting mechanism
FEP	front end processor
FOP	frame operating procedure
FSN	frame sequence number
IP	internet protocol
LAN	local area network
LEO	low Earth orbit
OMG	object management group
OMG	object management group
PIM	platform-specific model
PLUTO	procedure language for users in test and operations
PTV	pre-telemetry verification
SCL	spacecraft command language
SDTF	Space Domain Task Force
SOLM	satellite operations language meta-model
SPELL	satellite procedure execution language and library
STOL	spacecraft test and operations language
TCL/TK	tool command langue/tool kit
TDM	time division-mutliplexed
TDM	time division multiplied
XML	extensible markup language
XTCE	telemetric and command exchange

Chapter 7
Mission Data Processing and Distribution

Marilyn K. Dubas
Software Engineering Subdivision
Computers and Software Division

7.1 Introduction/Background

Mission data processing and distribution (MDP&D) is the portion of the ground segment encompassing the functionality that directly supports the purpose of a space system; the reason it was acquired and is operated, often referred to as the mission. Mission data flows from space to users, usually to multiple entities that use the data, and is transformed and often augmented during this flow. This chapter will discuss the mission data chain (MDC) that includes all the functions/activities related to the mission data and information from the time it leaves the space asset until products reach the ultimate users. A primary user may be (as in navigation systems) the space system's own ground system which utilizes the mission data to provide feedback to the space segment, usually via the asset command and control function. Figure 7-1 illustrates MDP&D within the ground segment reference architecture.

Space systems tend to fall into distinct mission categories such as remote sensing, navigation, communications, transportation, and space servicing. Mission data, information, and products vary based on the space system mission category. Remote sensing systems (e.g. Defense Meterological Satellite Program [DMSP], Geostationary Operational Environmental Satellite [GOES], Mars Rover) have sensor payloads which generate information as a result of sensor detection and/or activation. Communication payloads receive and retransmit communications (e.g. Tracking and Data Relay Satellite System, International Maritime Satellite). Navigation systems (e.g. global positioning system [GPS], Galileo) generate and transmit clock and position information. Transportation systems such as the Soyuz or the SpaceX Dragon provide information on the status of what they are transporting. Space servicing systems such as Orbital Express provide information on the space systems being remotely serviced. Aspects of the mission data chain are easily identified in remote sensing systems, but are present to some extent in all mission categories.

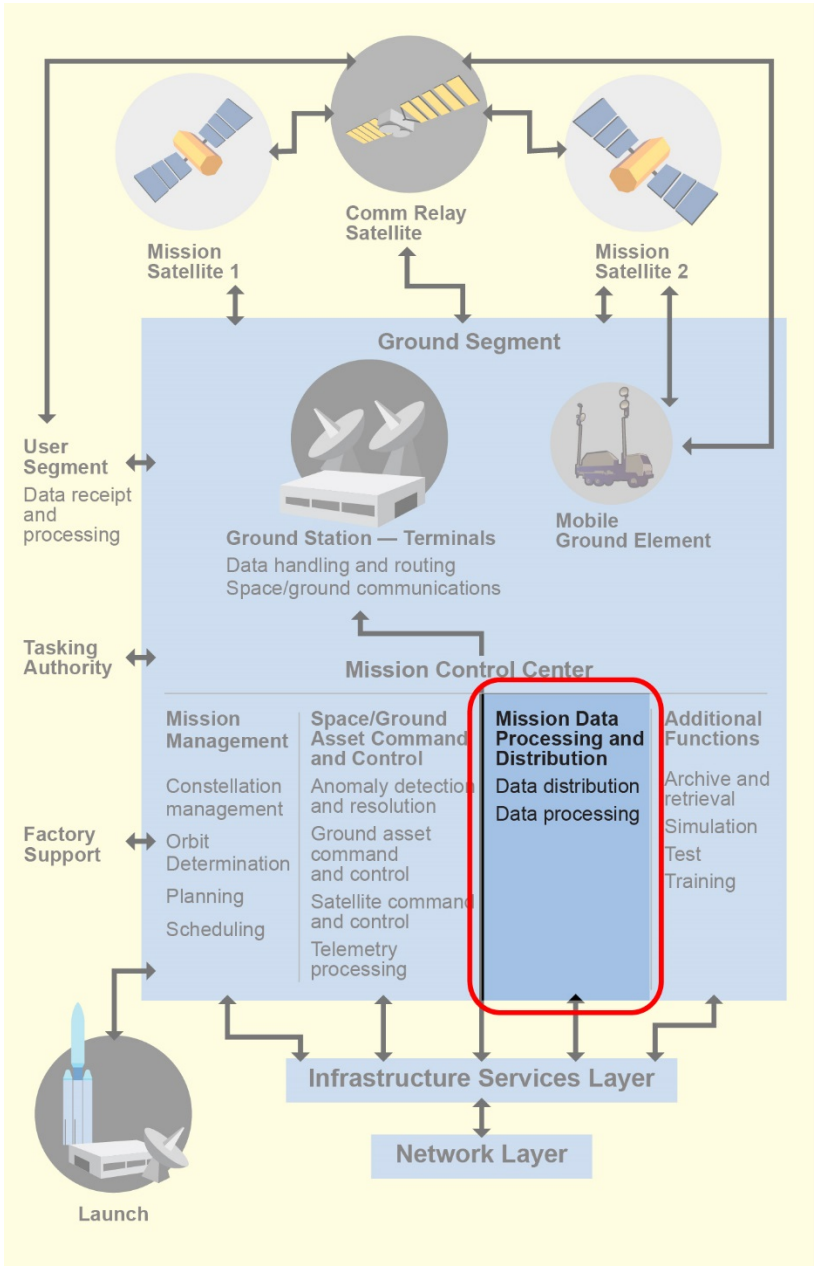


Figure 7-1. Mission data processing and distribution within ground segment reference architecture.

7.2 Definitions

Dynamic range Range of values a parameter can take on

Radio frequency (RF) Electromagnetic radiation in the range of 3 Hz to 300 GHz

Space asset Space segment entity associated with a ground system. Usually a satellite (bus plus payloads), but can also refer to hosted payloads.

7.3 Detailed Description

The functionality encompassed by the mission data chain can be grouped in a variety of ways, hence there is no standard way one will find it identified in individual ground systems. The front end of the mission data flow/chain is often referred to as mission processing. Many of these functions operate in "real time" as RF energy is transformed into digital information and initial mission products. Mission processing functional areas include data capture, data conditioning and preprocessing, and mission product generation (Figure 7-2). Data product distribution may take place from near real time, such as missile warning messages, to the provision of products from ancient archives, such as those provided by Landsat. Functional areas included in product distribution are exploitation and dissemination.

7.3.1 Mission Processing

7.3.1.1 Data Capture

Data capture is the first step in the mission data flow and/or chain and the conversion of the space-to-ground communication mechanism (RF in most current space systems) to a form usable on the ground, primarily to digital formats. A few ground systems utilize analog formats. Data at this stage can be archived and reprocessed at a later time. Data capture takes place in ground terminals. Control of terminal hardware, including antenna configuration and pointing, supports data capture. RF to digital sub-functions accomplished in terminal hardware includes: demodulation, demultiplexing, analog-to-digital conversion, bit synchronization, switching, decoding, decryption, and decompression.

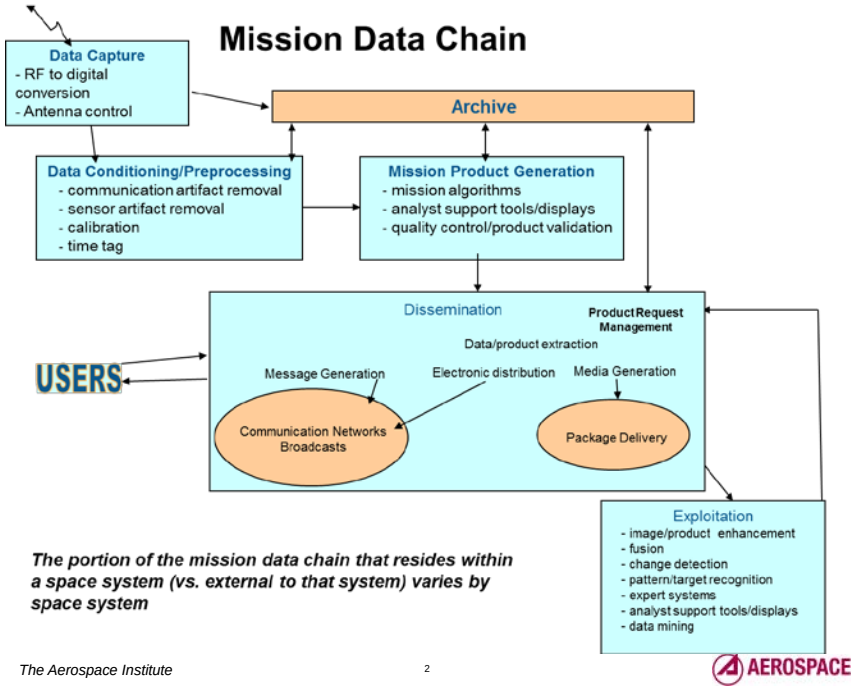


Figure 7-2. Mission processing and distribution mission data chain.

7.3.1.2 Data Conditioning and Preprocessing

Once mission data is in a useable format, steps are taken to prepare the data for product generation. Processing to eliminate and/or compensate for known and/or detectable flaws and non-uniformities is applied. Time is associated with the data. Once preprocessed, data is often archived.

Errors can be introduced by the space-to-ground communications and data capture. Bit error rates are usually very small and upper limits may be specified by systems requirements. Bit error rate requirements influence the design of the space-to-ground communications and ground terminal design. Commonly utilized space-to-ground communications formats (such as CCSDS) include error detection and error correction capabilities. These are applied to data to correct what can be corrected and to flag data that has been identified to contain errors.

The space hardware involved in the generation of mission data can have known flaws or conditions that affect the quality of the mission data produced. Such artifacts are identified during ground testing of the space hardware and continued testing once it is on-orbit. Compensation and accommodation for

identified artifacts is done during preprocessing. For example, dead and bad pixels in a sensor array are artifacts that are identified for later mission processing algorithms.

Space hardware involved in the generation of mission data differs by mission category. However, all hardware has parameters that need to be accurately estimated to properly interpret the mission data generated by the hardware. The calibration function in preprocessing consists of applying adjustments to the mission data to compensate for the particular characteristics of space hardware, which was previously determined by testing and analysis. For example, most remote sensing systems will have multiple channels of data that need to be balanced with respect to one another and approximately scaled in dynamic range. Navigation systems generally require calibration to account for space clock characteristics and RF delay characteristics in the space segment signal paths. The testing and analysis necessary to establish a calibration database will normally be performed on the ground before launch and in special on-orbit operations after launch. The important environmental factors and conditions are usually better controlled and understood on the ground. Initial on-orbit operations will usually include extensive calibration data collection. To account for changes over time, calibration operations are scheduled periodically during the mission life of the space asset.

In many space systems it is important to associate a ground-based time with mission data. This can be true even when data has space-generated time associated with it. Mission and/or system requirements specify the accuracy and frequency of "time tags." Ground systems include sophisticated atomic clocking systems often tied to the national timing system.

7.3.1.3 Mission Product Generation

Mission product generation varies by mission categories. For communication systems, the data capture and preprocessing functions alone usually recreate the data originally transmitted to the space asset and then relayed to the ground, so no additional processing is required. For other mission categories additional processing is usually required.

Mission product generation turns the data received from the space asset into something useful, i.e. mission products. This transformation is accomplished utilizing algorithms (processing techniques). Many algorithms may be involved for a single product type, each requiring many years of research and development. Multiple products may be generated from the mission data. Depending on requirements, these separate products may be generated simultaneously through parallel processing paths and/or strings. Timelines for generation of products vary and are driven by mission requirements.

Payload developers are often responsible for the development of initial algorithms as they are often the most knowledgeable experts on the mission hardware. The payload developer demonstrates the viability of their processing concepts/approaches, at least in a theoretical environment, with prototype processing capabilities. This may be referred to as the payload factory environment.

Although the payload developer is responsible for the science behind the algorithms, the payload contractor may not be the contractor that implements the algorithms in the actual mission processing system(s). In such cases the transfer of the "wisdom" from the algorithm developer to the processing implementer is a challenge; sufficient documentation is required to be produced by the algorithm developers and is transferable to the implementers.

In addition to pure mission data, mission algorithms may require space asset status information, which can be relayed to the ground in telemetry streams separate from the mission data. If not also embedded in the mission telemetry, this space asset status data must be provided to the mission processing system by the ground system. For space systems that support and/or include user elements all the necessary processing information must be available in the mission telemetry stream. Space asset attitude and ephemeris information is often necessary for mission processing algorithms. Very sophisticated subsystems (often taking years of development) are included in ground systems to generate accurate attitude and ephemeris estimates for mission processing. In some systems mission tasking information is also utilized in the mission processing system.

Over time space assets age and components may fail. Mission processing may still be able to produce valuable mission products if algorithms are modified or new algorithms are introduced to accommodate the space asset changes. It is not uncommon that over time improvements are made to processing techniques and additional product types are developed. Additional and modified processing capabilities are included in mission processing, either within the space system and/or externally.

Common in academic circles, NASA, NOAA, and European remote sensing systems is the concept of data levels that range from very raw data to highly processed information. The mission data levels are defined as follows:

- **Level 0—**Reconstructed, unprocessed instrument payload data at full resolution; any and all communications artifacts (e.g., synchronization frames, communications headers, duplicate data removed).
- **Level 1A—**Reconstructed, unprocessed instrument payload data at full resolution, time referenced, and annotated with ancillary information, including radiometric and geometric calibration coefficients and

georeferencing parameters (e.g., platform ephemeris computed and appended but not applied to the Level 0 data).

- **Level 1B—**Level 1A data that have been processed to sensor units (not all instruments will have a Level 1B equivalent).
- **Level 2—**Derived geophysical variables at the same resolution and location as Level 1 source data.
- **Level 3—**Derived geophysical variables mapped on uniform space-time grid scales, usually with some completeness and consistency.
- **Level 4—**Model output or results from analyses of lower level data (e.g., variables derived from multiple measurements).

Most product generation systems include human operators that require information and possibly mission products. This operator involvement can be as simple as assuring the processing system is operating as expected and required. However more mission-oriented involvement is often the case. Operators may be required to make judgments in order to develop mission products. For example, an operator may need to release messages based on mission data content presented as visual or audible information generated by mission processing. What operators use and what they cause to be generated can all be considered mission products. The operators usually need specially developed tools to manipulate the mission data and products, and additionally require proper presentation mechanisms to perform analysis and arrive at conclusions. These tools and analysis capabilities may be what generates the finished products.

Quality control and product validation associated with mission processing are important processes in the generation of mission products. These functions can be off-line activities perhaps done by organizations separate from "real-time operations." These functions may be implemented through representative sampling and analysis of products associated with the operational processing strings. For example, imaging systems such as Landsat include systematic, quality-control measures.

7.3.2 Distribution

Later stages of the mission data chain include dissemination and exploitation.

7.3.2.1 Dissemination

Dissemination supports both the first-time flow through the mission data chain as well as later iterative cycles of data and/or product retrieval, reprocessing, exploitation, and distribution to users. Dissemination can be anything from short "real time" electronic messages, to a physical item such as an image delivered

on paper. In our electronic age fewer physical products are being distributed by ground system elements. Physical product generation is being delegated to users.

Data may be distributed by a variety of mechanisms, to include: network connections; placing the data in an accessible repository; or point-to-point communication links. Most space systems are moving away from using point-to-point connections in favor of networking. Some connections will remain on point-to-point links for special situations due to security or policy considerations. For such actual point-to-point transport of data and products, dissemination utilizes existing communications infrastructure such as Defense Information Services Agency (DISA) networks, satellite communication (SATCOM) and commercial fiber providers. Internal to ground systems is the functionality to interface with such national communications resources.

For some systems, there is no sharp distinction between the dissemination and archive functions. Increasingly, mission data dissemination is accomplished by placing the data in a repository accessible to authorized users. This repository may be considered part of the space system, or it may be considered an external user system. This difference can be true for both the physical location and/or ownership of the repository, as well as for the logical custody of the data in the repository. In fact, the answer could be different for the physical and logical views of the same repository. This can lead to ambiguous roles and responsibilities when issues arise.

In near-real-time scenarios, product from mission processing is received and immediately prepared for distribution via communications systems. Preparation may include formatting according to communication system standards and/or using community standards. Management of data and products is included in dissemination functionality. This includes the ability to receive and respond to requests for data and products from users. User product requests can result in tasking for exploitation. Extraction of data and products from repositories is included in a dissemination functionality. Some distribution infrastructures include user support and/or help desk capabilities staffed by operators to assist the using community. Extensive work has been done by stakeholders to arrive at standard formats for data and product distribution. Older programs are not always compliant with such standards.

Metadata which provides descriptive, context, and summary information for a dataset and/or product is often generated and associated with its “parent” in archives and distribution systems. Repository search capabilities are based on the metadata content.

Systems which include reprocessing of data are required to maintain not only the data but also all of the supporting information required to process the data, including information from sources external to the mission data chain.

7.3.2.2 Exploitation

The purpose of exploitation activities is to refine existing products and to discern additional information from mission data. For this discussion of the mission data chain, product generation refers to direct processing of a mission data stream executed as the data arrives in the ground system elements without significant delay, i.e. "near real time." Exploitation refers to later processing and reprocessing of mission data and products, often utilizing multiple sources. This is accomplished through the use of special processing techniques, such as those identified in Figure 7-2, and combining multiple sources. Often over time new information content is discovered which prompts the development of additional exploitation processing techniques and new products.

Exploitation often takes place in centers devoted to particular specialties, such as weather processing. Disparate organizations develop and operate such centers. Processing in these centers includes anything from starting from scratch by reprocessing raw mission data with alternate processing techniques, to analysis with specialized tools and generation of new fused products. Exploitation processing normally involves operators who make judgments based on the information provided by their tool set. However, automated processing can also be included once the merits of the techniques and resulting products are established.

Historically certain communities have used alternate breakdowns of functionality especially in remote sensing systems. Two such alternates are presented in Figure 7-3: tasking, collection, processing, exploitation, dissemination (TCPED); and tasking, posting, processing, using (TPPU). The tasking functionality in both alternate approaches is part of the mission management functionality.

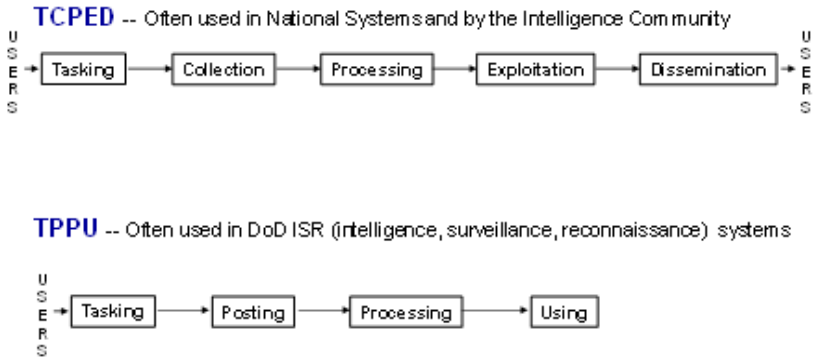


Figure 7-3. Alternate functional breakdowns.

7.4 Technical Considerations

Mission data volume and mission product timelines, quality, and classification levels pose technical challenges for MDP&D systems. In most space systems, the space communication system is the limiting factor on the amount and quality of mission data that reaches the ground. The space mission hardware is often capable of generating more mission data than the space-to-ground communication links can accommodate. The down sampling and/or compression of the on-board generated mission data is a space system-level design consideration. However the ability to send more mission data to the ground is ever expanding. On-board hardware is generating more data, spacecraft can transmit data a higher data rates, and ground communications bandwidth is increasing. These trends must be accommodated by mission processing and distribution hardware and software.

Product timelines bring additional challenges for MDP&D. A processing architecture to generate products in a short time (e.g. seconds) from large quantities of data received at high data rates may be difficult to design. Tailored and/or specialized approaches may need to be developed. In addition, product size likely increases with mission data volume, which in turn affects distribution architectures. The length of time that data and products must be maintained in repositories is another architecture driver.

Systems may have to deal with multiple classification levels. Even when the mission processing can operate at only one classification level, data and products may need to be distributed to users at a variety of classification levels. The dissemination functionality must take into account constraints on how users can access the data (e.g., the MDP&D element may produce different types of products for different types of users, and some users may not be allowed to access some types of data). Typically, the dissemination function must enforce these constraints, supported by some form of identity management for user accounts.

Mission processing frameworks are being developed that facilitate the flow of mission processing steps in a non-mission specific fashion. Mission-unique software to implement various processing steps is accommodated. Multiple approaches to various processing steps can also be accommodated by the frameworks. In the future such government-off-the shelf/commercial-off-the shelf open architectures may be able to be used as the basis for MDP&D systems. One example is the Utah State University Research Foundation's Space Dynamics Laboratory, in collaboration with the Naval Research Laboratory, development of VANTAGE, a tactical imagery exploitation framework. VANTAGE is free for government use. Classified frameworks have also been developed.

7.5 Programmatic Considerations

The portion of the mission data chain that is considered to be part of a space system, compared to being separate and external, can vary greatly. Even though it may be recognized as part of the total space system, some or all of the ground system may be acquired separately from the space segment and/or asset. The work of different contractors on separate contracts must be integrated together with coordinated delivery schedules to support the launch and operations of space systems. This is especially true in the case of mission processing systems. Also common is that various agencies and/or organizations develop unique systems to address mission processing, exploitation, and dissemination of products. For example, weather and/or meteorological programs have multiple agencies that procure unique user elements. Often multiple exploitation centers exist and utilize data and products from multiple space systems. Multiple dissemination systems may provide data and products to such centers. Such exploitation centers and dissemination systems are procured separately from space systems and from each other. Maintaining compatibility is an ongoing technical and programmatic challenge.

7.6 Bibliography

Benator, Sheri, et al., *Ground Systems for Satellite Operations: Primer and Acquisition Considerations*, Ground Systems Architecture Workshop Tutorial. The Aerospace Corporation, El Segundo, CA. March 27, 2006.

7.7 Acronyms

CCSDS	consultative committee for space data systems
DISA	Defense Information Services Agency.
DMSP	Defense Meteorological Satellite Program
GOES	geostationary operational environmental satellite
GPS	global positioning system
ISR	intelligence, surveillance, reconnaissance
landsat	land satellites
MDC	mission data chain
MDP&D	mission data processing and distribution
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
RF	radio frequency
satcom	satellite communication
TCPED	tasking, collection, processing, exploitation, dissemination
TPPU	tasking, posting, processing, using

Chapter 8
Additional Functions

Scott C. Wilkes
Software Systems Assessment
Software Systems Assurance Department
Dana Honeycutt
Technical Training and Development Department
The Aerospace Institute

8.1 Introduction/Background

Additional functions enable the primary capabilities of the ground segment, which include mission management, space/ground asset command and control, and mission data processing. There are a number of enablers, additional functions that support these capabilities. Although each enabler is important in the overall operation of a ground system, the main enablers are data archiving and retrieval, simulation, test, and training.

8.2 Definitions

Archiving and retrieval the process of moving data to a separate data storage device for long-term retention. Retrieval of archived data supports activities such as anomaly resolution and system state-of-health (SOH) assessment.

Simulation the imitation of a real-world system or process over time.

Test executing a pre-planned sequence of actions and capturing the results to verify that a ground system meets requirements and is behaving as expected.

Training the orientation of the end-user and the system maintenance and sustainment team to all aspects of system design and operations.

8.3 Additional Functions Overview

The primary ground segment capabilities are: the mission management subsystem which plans and schedules the space system; the space/ground asset command and control subsystem which commands, controls, and monitors space assets and related ground equipment; and the mission data processing and distribution subsystem which processes raw data collected by the space asset payload and distributes the processed data products to end users. Supporting these capabilities are eleven enablers: archiving and retrieval, simulation, test, training, communications, maintenance, facility management, administrative services, user services, information infrastructure, and security. Although each

enabler is important in the overall operation of a ground system, this chapter will limit discussion to the main enablers: data archiving and retrieval, simulation, test, and training, as shown in Figure 8-1. Additional functions are primarily supporting functions that enable or assist ground system development and operations.

Archive and retrieval stores years of down-linked telemetry frames and payload data that can be retrieved and analyzed to support state of health (SOH) assessment for the mission management subsystem and anomaly resolution for the space/ground asset command and control subsystem. Logging records, operator entries, and system operations events generate large amounts of data that is also archived for later retrieval to facilitate monitoring and troubleshooting during system maintenance. Data archives contain data that is retained for future reference by users and maintainers, either to satisfy requirements or for regulatory compliance. Data archives are indexed and have fully-featured search capabilities so that specific data items or groups of data items can be easily located. A fully realized retrieval capability is advantageous due to the large amount of data that may be stored over the mission life of a satellite or the even longer lifetime of the ground system spanning multiple satellite lifetimes.

Simulation is used during ground system development for integration and test, verification, and validation activities. Simulations are used to imitate portions of the space and ground system and environment in order to exercise and enable the testing of the ground system prior to operational deployment. Simulation is also used during operations to test command sequences prior to upload to ensure that the commands execute correctly and have the desired effect. Simulators can assume several forms, from a pure software simulator simulating ground system interfaces to a hardware-in-the-loop (HITL) simulator that provides high-fidelity satellite simulation. After the system is delivered and is operational, simulation supports the space and ground asset command and control subsystem as a pre-upload command sequence verifier. Simulation also supports the additional function of training. During hands-on training, a satellite simulator is used to provide realistic responses to trainee inputs.

Test is used throughout ground system development as the software is unit-tested and the hardware and software configuration items (CIs) are integrated into subsystems and elements. Testing is performed at several points during the ground system lifecycle: system development, system qualification, post-system delivery, and during maintenance. In the context of additional functions, the test drivers, data set scenarios, and special test equipment supporting the execution of formal and informal testing are included. When development is complete the ground system requirements are verified through test, and user scenarios are

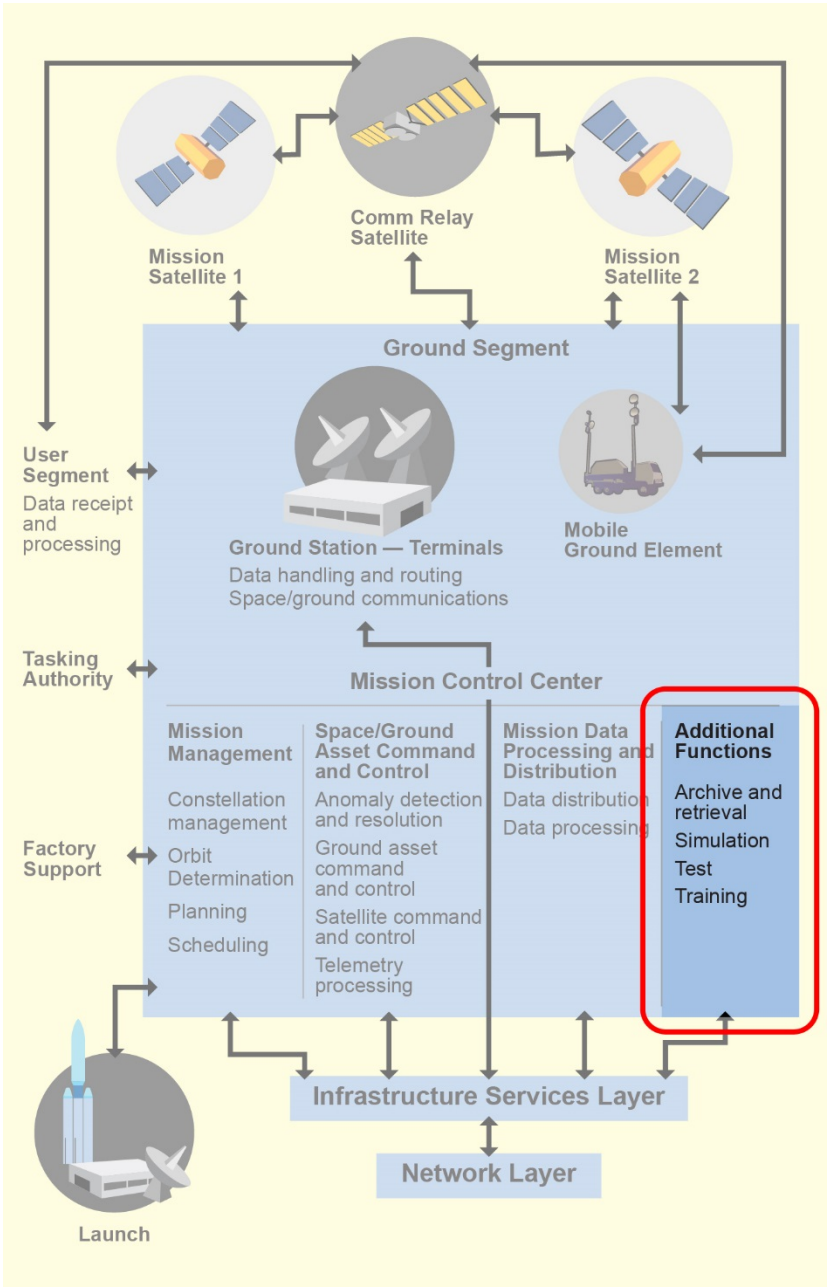


Figure 8-1. Ground segment reference architecture additional functions.

validated at the factory and on-site through test within the target environment. Regression testing is performed during system maintenance to ensure that software and hardware changes do not introduce unintended side effects on other parts of the system. Each of these phases requires several types of external drivers, special test equipment, and scenario data to enable the testing when the real interfaces and data are unavailable.

Training modules, materials, and exercises are developed starting soon after the critical design is completed and are taught as classes by the ground system development team. Training and qualification of operators and maintainers for a complex system can cover several weeks to explore each aspect in depth. Training is used to acclimate operators and maintainers to the system user interface and system design. Training occurs before and as a system is fielded so that operators and maintainers are prepared for the transition to operations. Operator training focuses on typical responses to command and control inputs and error conditions to support the mission management function. Likewise, training for maintainers has a more in-depth design and implementation orientation to enable the factory support maintenance team. Launch training is provided for the specialized launch team and refreshed prior to each launch campaign.

8.3.1 Archiving and Retrieval

The archiving and retrieval function is accomplished by software and database applications interfacing with dedicated hardware storage devices used to store the long-term data. The software calls the database primitives to store the data to and retrieve the data from the archive hardware. The data is then formatted for display to the user and/or written to a file.

The size and type of hardware storage device used in a particular ground system is dependent on the amount of data to be stored and the length of time it needs to be available for retrieval. Performance requirements will specify the expected data accumulation rates and the length of time that the data must be kept in long-term storage. These parameters, plus the required capacity margins, dictate the amount of archival storage that must be initially purchased and integrated. In addition to gross size requirements, other archive and retrieval requirements will specify input data format, required retrieval speed, and data longevity. As a typical example, if the requirements specify that the raw payload data, received at 300 Megabits/sec., must be archived for 90 days, then the size of the archive can run in the terabyte range. Storing telemetry frames is another massive storage space requirement, since frames are received several times a second and they are typically required to be kept for a year or longer to support anomaly resolution. Modern archival storage technology enables large data capacities as the cost per gigabyte has decreased so much that multi-terabyte archives are now commonplace.

Occasionally, data that is outside the current lifecycle of the on-orbit asset or ground system will need to be accessed as part of anomaly resolution efforts or historical trending. It is important to ensure that older data, whose storage and retrieval mechanism may be out-of-date (particularly with a ground system that has been in place for a long time) can still be accessed as the ground system infrastructure is upgraded. This is important since migrating large amounts of data from older storage mechanisms to newer technology may not be cost-effective.

Archives are usually divided into two functional parts, an online archive and an offline archive. The online archive is typically a 30 day archive with faster retrieval and data transfer rates to enable quicker response to requests for current data. Each day, the oldest data (Day 1 is newest and Day 30 is oldest) is moved from the online archive to the offline archive and the current day is written to the front of the queue (new Day 1). The offline archive is much larger than the online archive and has lower retrieval and data transfer performance. Some cost savings are realized by this approach due to the lower cost of moderate-performance, high-capacity, bulk storage hardware. In this case, retrieval performance is traded off against the lower frequency of requests for older data. The archive and retrieval software keeps track of which archive contains which data so that the archive architecture is transparent to the archive and retrieval consumer.

There are many uses for the archived telemetry, command, mission, and log data. Modern telemetry, tracking, and command (TT&C) systems have sophisticated functions that enable a range of telemetry frames to be played back in the order they originated from the satellite. Telemetry measurands can be monitored, trended, and plotted just as if they were being transmitted in real-time from space. Retrieved command strings can be played back and fed into a satellite simulator to confirm or troubleshoot observed satellite responses. Mission data can be played back and reprocessed using different parameters to improve upon previous processing or to extract different information. Satellite contractors access and analyze current and long-term data to help improve designs on new satellites. Logs can be retrieved to enable analysis of operator actions as well as anomalous ground system behavior not necessarily associated with a satellite.

Archives can also be exploited in non-traditional ways. Recent advances in data mining are bringing new techniques of extracting meaningful information from large data sets. Applying data mining analytic principles to large amounts of data, such as that contained in a ground system archive, can reveal consistent patterns and/or systematic relationships between variables. The findings can then be validated by applying the detected patterns to new incoming data in search of continued correlations. These results can enable second-tier analysis

seeking to improve the space-ground system or resolve persistent anomalies not solvable by traditional means.

8.3.2 Simulation

Ground systems require multiple simulators in order to fully integrate and test the ground system during development and operations. The most common is a satellite simulator which simulates the responses of the satellite to ground system commands. Other simulators are written by developers to provide simulated interfaces or subsystem functions to enable stand alone testing during development. Simulators are also written to simulate ground hardware such as front-end processors (FEPs), encryption units, and ground antennas and radio frequency (RF) equipment chains when that hardware is unavailable at the development location.

Satellite simulators receive commands from the ground system and behave as the satellite would under normal conditions, including returning the appropriate telemetry data. The simulator can also generate off-nominal responses to test the ground system behavior under those conditions as well. Satellite simulators may be software-only or may be a hardware-software hybrid known as a HITL simulator that includes some actual satellite hardware to provide a higher fidelity simulation.

In a satellite simulator, the major bus subsystems such as attitude determination and control (ADCS), command and data handling, propulsion, power, and thermal subsystems are simulated. Most satellite simulators are known as bus simulators, which means that only the satellite infrastructure (the “bus”) and not the payload is simulated. Each subsystem is modelled using algorithms that have been derived from theoretical and real-world performance. Battery drainage curves, thermal gradients, and propellant tank levels are examples of simulated subsystem parameters that can be set as initial simulation conditions.

A satellite simulation starts with the satellite in a particular x, y, z location. For geosynchronous orbits, this location is relatively stationary and assumes a small figure-8 pattern about the equatorial geostationary point. For low Earth orbits (LEO) or highly elliptical orbits (HEO) this location is a starting point along an orbit ephemeris that is programmed into the simulation. With the simulation underway, the simulator’s modelled subsystems react in real-time to ground commands and the satellite’s position relative to the sun and the Earth. Delays are built into the simulator’s command and telemetry response times to represent the distance of the satellite from the ground antenna to increase realism. Responses to the commands and changing position are reflected in the simulated telemetry sent back to the ground system. The subsystems dynamically react, for example, when the sun is in eclipse: the solar cells are dark, power is drawn from the batteries, and the thermal model shows cooling effects.

Another application of satellite simulators involves supporting the launch team. The simulator has the capability for injecting system issues and failures to mimic a system anomaly or error conditions. This is especially useful during pre-launch exercises and rehearsals to orient the launch team with an anomaly resolution discipline. The failure injection effects are reflected in the simulated telemetry that is observed by the satellite engineers monitoring their respective subsystems. Once detected, an anomaly is declared and the resolution process is followed until the simulated anomaly is resolved, just as it would be during an actual mission.

Simulation is a key part of standard system development and aids greatly in the ability to design, develop, and test the system as it will be operated. Other types of developmental simulations emulate external interfaces so that ground software can test compliance with the appropriate interface control documents (ICDs) and demonstrate operation as intended. Ground subsystems developers may create rudimentary simulators for testing the internal interfaces with other subsystems. For systems that include mission data processing, a stand alone payload simulator can be developed to provide realistic scenarios of raw payload sensor data to test ground processing. These simulators can be very complex, depending on the nature of the sensor and the data model needed to simulate it with sufficient fidelity to accurately test the ground processing algorithms.

Simulators can be built to test specialized or expensive ground equipment not available to the developer. FEPs, cryptographic encoder/decoders, and RF equipment chains are examples of equipment that may merit their own simulators to reduce development risk. For hardware-intensive ground systems, field programmable gate arrays (FPGAs) are frequently used when processing speed and power is paramount. FPGAs can be simulated in software to provide realistic responses for development purposes before the design is committed to silicon. Operations and support consoles providing the primary interface with operational and support personnel can be simulated to provide representative layouts for testing and fine tuning human-system interface (HSI) display interactions.

8.3.3 Test

The test function provides the test drivers, scenarios, and special test equipment needed to help establish a realistic end-to-end test environment required for verification, validation, certification testing, and other special tests. In ground systems development there is testing known as factory acceptance test (FAT) that is focused on verification of requirements by the developer at the factory. During transition to operations, there is post-installation testing at the site known as a site acceptance test (SAT). Validation testing is another type of testing, which differs from verification testing in that functions and capabilities are

exercised using day-in-the-life scenarios to ensure that the system fulfills its intended use in addition to satisfying its requirements.

The test-like-you-fly (TLYF) approach is accomplished using simulators, test drivers, test data sets, and special test equipment to create a realistic test environment that provides the stimuli needed for TLYF testing. These test enablers are commonly aggregated into an integrated testbed that is used for ground system testing during development and when target sites, operational complexes, or other operational support areas are not available.

Test drivers are software applications that provide the initial conditions for a test and stimulate a subset of the system such as a subsystem or configuration item (CI) to enable focused testing. Individual developers have differing needs when running tests and frequently do not want to incur the overhead required to bring up the entire system when a quick, targeted run of a subsystem or CI will provide the needed results. Development resources are typically in high demand and the faster a test can be run and the results be obtained, the faster the resources will be available to the next developer. Commercial products that permit automation of entire test sequences also fall under the test driver category. These products use advanced scripting to provide input stimuli and capture output responses for post-test review. By using an automated test driver tool, tests can be queued up to run overnight when system resources are not being utilized, or being under-utilized, with the results available when the developer returns the next morning.

Scenarios used for testing can come in several different forms. Day-in-the-life scenarios are created using both nominal and off-nominal operator actions. These actions are scripted into sequences that operators use to exercise the system in a realistic way by emulating actual operations. These scenarios enable validation of the system implementation and the system operations concept. Off-nominal scenarios are especially important as they use less-frequently tested paths and exercise the developer's error-handling implementation and the overall system robustness. Scenarios can also be data sets that are used as inputs to test mission data processing algorithms and functionality. Data sets such as sensor payload data can be provided by ground system customers or generated through a payload simulator. These data sets will have both raw sensor data and metadata, such as date and time, satellite orbital ephemeris, and sensor parameters at the time the data was captured. Special data sets for satellite simulators (discussed previously) that provide the initial conditions for a particular satellite position, orientation, and SOH also fall under this category.

Special test equipment (STE) is hardware that is used to monitor, measure, simulate, or troubleshoot ground functions during development and testing. Hardware that is difficult to simulate, such as cryptographic equipment, can be purchased and integrated into the testbed as STE. Equipment such as network

analyzers are connected to interfaces and buses to visually display activity on the interface or bus. Oscilloscopes and spectrum analyzers are connected to RF outputs to visually verify that the ground station equipment is generating signals and operating as expected. Modern network analyzers, oscilloscopes, and spectrum analyzers have storage functions so that time slices of activity can be captured and displayed or printed for off-line analysis. STE is also used in the satellite factory to connect to satellite hardware under development. In this way, a ground system can be connected to the satellite and TT&C functions can be checked out prior to satellite pre-delivery testing.

8.3.4 Training

Training the operations and maintenance (O&M) personnel is an important part of the system installation and delivery activities. Operator training is essential to maximize the efficient use of ground system capabilities and tools. Depending on the terms of the O&M contract, the operators can be developer personnel who are already familiar with the system, perhaps as software or test engineers, or military and/or government personnel who are brand new to the system. This wide range of pre-training skillset means training needs can vary greatly with respect to the depth and breadth of information required and should be accounted for in the training plan. The other half of O&M, maintenance, is a separate category of personnel requiring training. Maintenance training differs from operator training in that the course materials include review of system development specifications, design materials, and formal test results in addition to much of the operator training material. A complex ground system can easily take multiple weeks of classroom training to cover all functionality. At the end of the training, certification examinations are administered to ensure that the operators and maintainers are ready for the transition to operations.

Development of training course materials is begun mid-way through the ground system development, typically after the critical design review (CDR). Although the HSI is designed to make many of the system operations intuitive, some amount of formal training is necessary to ensure user understanding of all of the possible menu options available on a complex system. These training materials typically present an in-depth overview of the system before walking through the major functions. Screen shots of menus and graphical user interfaces showing realistic data displays enhance the materials. Command procedures, or PROCs, are sequences of satellite commands stored in files and uplinked to perform specific functions. Each PROC is covered in the training, examining input options and execution effects on satellite operation.

Maintenance includes system administration, configuration management, installation and regression testing, and discrepancy troubleshooting. Maintenance is divided into echelons: Echelon 1 is the on-site maintenance team; and Echelon 2 is the factory maintenance team. The Echelon 1 team is the

maintenance front-line and interacts with users on a daily basis. A responsive Echelon 1 goes a long way towards maximizing user satisfaction with the new system. This responsiveness is enabled through classroom and on-the-job training. Each maintenance aspect is covered in detail; from adding new users to installing software patches, to regression testing, to swapping out faulty equipment with spares and arranging for repair or spare replacement. Simulator operation and maintenance rounds out the curriculum.

Supplementing classroom presentations are hands-on exercises. For The Space and Missile Systems Center (SMC), the training programs are developed on the Standard Space Trainer (SST) framework developed by Sonalysts. This framework provides the tools and infrastructure to develop a stand-alone, computer-based training system for any operational ground system. The SST allows individuals or full crews to exercise operational scenarios with instructor guidance and to be evaluated by an instructor against training objectives.

In addition to normal operator training, special training exercises can be scripted to prepare the operations crew to support events such as launch and anomaly identification or mission-oriented operational exercises involving multiple systems or agencies. Recent trends involve providing early training and then allowing operator participation in integration and test through running test procedures such as those for FAT and SAT. This early hands-on training is proven to accelerate system familiarization, and as a bonus, is a no-cost augmentation of developer staff to cover test shifts when running around the clock.

Many organizations pursue the “train the trainer” approach where an initial set of operators and maintainers are trained by the ground system developer and then are provided with the course materials. That set of personnel then trains the remaining operators and maintainers in turn. Videotaping of the initial training sessions is also popular and provides another option for delivering the material to remote sites. Having the course on video allows course review on-demand for knowledge refresh and also provides a way of training O&M personnel as the initial crew rotates out or experiences attrition.

8.4 Best Practices

8.4.1 Archival and Retrieval

Some best practices and lessons learned for the archive and retrieval function include:

- Review developer assumptions on the sizing of the data archival system and make sure that the assumptions and calculations are correct and

satisfy the archive requirements. Ensure that all data types to be archived are represented.

- Review the developer's archive operations concept, especially the short-term to long-term archive transfer concept and ensure that this meets needs as currently understood. Insist that the short-term to long-term threshold be programmable so that it can be varied if post-delivery experience dictates some adjustment.

- Investigate the existing archive infrastructure and understand the issues with interfacing; this allows contractor trade studies to be completely evaluated for ground systems replacement or upgrading existing systems. Delay buying archive hardware for the operational site as long as possible. The advance of technology ensures that it will be technologically obsolete soon after it is purchased. Delaying as long as possible ensures that the price per gigabyte is as low as possible.

- Know in advance the estimated cost to migrate the old data so that developer options can be evaluated in terms of the cost to migrate compared to performance of existing hardware.

8.4.2 Simulation

A major technical consideration with simulation is the qualification of simulation and other test tools for use in system tests. This is crucial because requirements verification that uses simulation as part of the test environment is dependent on the simulation being a faithful representation of the item being simulated. Simulators should have pedigree paper trails, indicating that the developer has run qualification tests against the simulator and compared the responses against the simulated real-world satellite or device. Simulators are maintained just like any other software or hardware and will have a database of problem reports. Programs should insist on getting regular updates to the problem report database and review the types of problems for ones that may affect the on-going development.

From a programmatic consideration these best practices should be followed:

- Ensure scenarios and data sets used for testing are supplied with the simulator.

- Simulators should be procured as early as possible so that they will be available for developers to use in their early and subsequent testing.

- Ensure the contract provides for receiving the maintenance drops and that continuing developer support is included. The developer frequently upgrades the simulator, either adding new features or fixing problem reports.

8.4.3 Test

The following are best practices when considering test:

- Advise the contractor to procure automated test tools. There is up-front cost and schedule overhead in automating a test but after the automation is complete it will pay for itself within a few tests and can be delivered to the O&M team for automated regression testing.

- Ensure that the developer has planned for and will implement multiple sequestered environments for operations, maintenance, and training. Being able to test and train while in operations without any activity affecting any other is essential to meet availability and maintainability quality factors. Put delivery of key developer tools and drivers on contract to supply to the maintenance team plus sufficient documentation to understand them. This will enable the maintenance team to more efficiently test and troubleshoot system issues during operations when system down-time can have real consequences.

- Ensure the developer identifies required STE early and gets it on order as soon as possible. Some specialized equipment can have long lead-times and not be available when needed. Some STE will be government-furnished equipment (GFE), which means the government program office will be responsible for providing it. Any delay in meeting the developer's need date for that STE will be leveraged by the developer in terms of cost and schedule relief. Investigate the cost of renting the equipment (if possible) if needed to bridge any gaps.

8.4.4 Training

The following are best practices addressing training:

- Ensure that use cases and concept of operations are up to date and reflect the real world. Direct the developer to establish realistic scenarios using these documents.

- Ensure that the developer plans to cross-train operators for more than one position. This increases crew flexibility and provides mental stimulation for what can be (on good days) a mundane job.

- Video record classes to ensure availability long after the developer has fulfilled their contractual training obligations.

- Direct the developer to write the O&M plan sooner rather than later. Having the plan in hand will inform on-site staffing levels so that personnel with the right kinds of backgrounds can be located without the time pressure of installation and transition to operations.

- Define with the developer the extent of early operator participation in test activities and ensure that it happens. It's a win-win with the developer getting free labor and the customer getting an early start on operator familiarization.

- Carefully review the training plan and ensure that the plan will meet operations and maintenance needs.

- Don't wait too long to start training materials development. These materials need to be in place and training started well before the final system is delivered.

8.5 References

1. Lutton, David. *Test Requirements for Ground Systems*. TR-2013-00215, The Aerospace Corporation, El Segundo, CA. 2013.

2. White, Julia D. and Lidnsay G. Tilney. *The Test Like You Fly Process Guide for Space, Launch, and Ground Systems*. TOR-2014-02537, The Aerospace Corporation, El Segundo, CA. July 2014.

3. AFSPC Guidance Memorandum (GM) 2014-13-01, *Space Operations Crew Force Management, Training, Standardization and Evaluation*. March 11, 2014.

8.6 Acronyms

ADCS	attitude determination and control
CDR	critical design review
CI	configuration item
F&P	front end processors
FAT	factory acceptance test
FPGA	field programmable gate arrays
GFE	government furnished equipment
HEO	highly elliptical orbit
HIS	human system interface

HITL	hardware-in-the-loop
ICD	interface control documents
LEO	low Earth orbit
O&M	operations and maintenance
PROC	command procedure
RF	radio frequency
SAT	site acceptance test
SOH	state of health
STE	special test equipment
TLYF	test like you fly
TT&C	telemetry, tracking, and command

Chapter 9
Infrastructure Services

Michael L. Campbell
Computer Applications and Assurance Subdivision
Computers and Software Division

9.1 Introduction/Background

The infrastructure services (IS) layer of typical national security space (NSS) ground segments includes the functionality assigned to the IS layer, the boundaries of the IS layer, and the interfaces on the boundary. Infrastructure services encompass all the hardware and software in a ground segment between the network layer and the mission-specific subsystems, as illustrated in Figure 9-1. The actual capabilities considered to be infrastructure have evolved significantly over time, and are currently undergoing a rapid pace of evolution due to the transition to cloud-based computing environments and service oriented architecture (SOA). In almost all cases, the infrastructure of a ground segment comprises at least the system health and status monitoring functions, and the interfaces for the internet protocol (IP) network-based communications and any specialized non-IP based communications. It may also include interfaces to facilities, such as for uninterruptable power supplies. Infrastructure for a NSS ground system depends on whether it is operational, under active development, or in acquisition planning.

Most legacy ground segments in the operation and maintenance (O&M), were acquired as purpose-built, hardware-software systems. Many of the older systems are implemented as stovepipes with no explicit infrastructure layer. More recent ground segments currently in O&M were still acquired as purpose-built, hardware-software systems, but are implemented as product line architectures, such as OS/COMET® and ECLIPSE®. In the product line approach, the contractor provides a set of pre-existing software components that can be reused (with or without modification) to implement some of the most commonly used ground segment functions, such as: a common operating system; inter-process communication functions, common interfaces to the local area network (LAN) and wide area network (WAN), computer resource monitoring and control functions, database management, metadata tagging, and time and frequency references, as well as higher-level functionality more specific to ground segments (e.g., orbitology; space vehicle contact scheduling; satellite command and control, telemetry processing, storage, and display). Some of the lower-level functions would be considered the infrastructure of these legacy ground segments currently in operations.

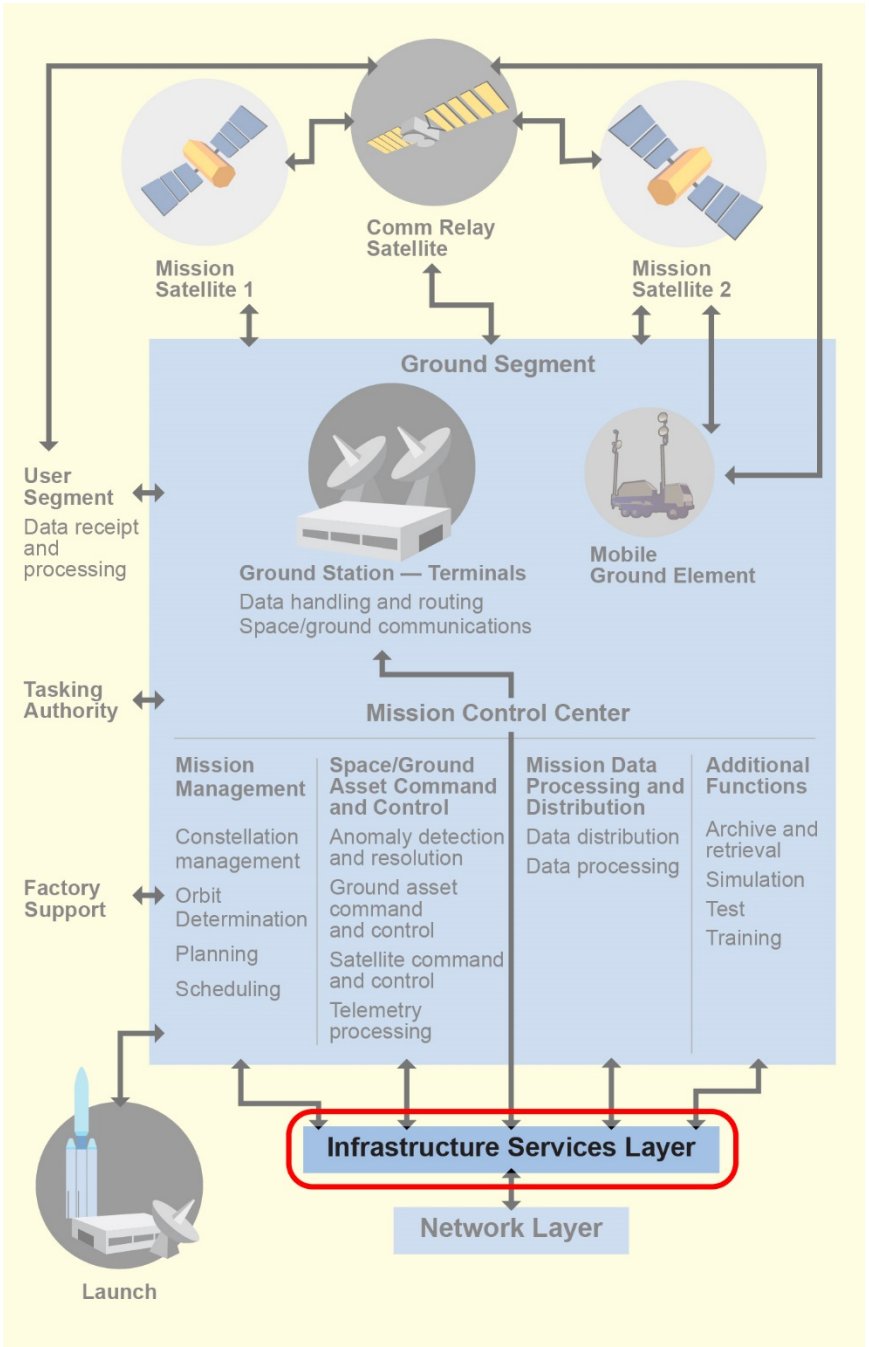


Figure 9-1. Insfrastructure services components.

Most NSS ground segments currently in development incorporate some form of layered approach where an infrastructure layer is called-out. For systems that have been under development for a significant amount of time, this may mean that a layered software stack is defined, which is similar across different physical hardware components of the system, and the software stack incorporates some infrastructure functions. Such layered systems provide, as part of the infrastructure layer of the software stack on each computer, commonly used functions such as those listed in the previous paragraph, as well as centralized health and status monitoring (ground segment hardware and software) and perhaps some frequently used security features such as access control and user account management. Ground segments developed relatively recently, possess an infrastructure layer where the overall hardware-software system is organized as a layered structure with a well-defined set of capabilities assigned to a discrete IS layer, and these common infrastructure capabilities are available to all of the mission-specific components. In these layered architectures, the IS layer typically provides all of the common functions, as well as additional functions such as a common database, features to support continuity of operations, and common security features that includes user identity management and key management. The processing hardware may be organized as a set of generic servers, rather than each machine being dedicated to a particular processing function. In this case, the collection of servers would typically be considered part of the IS. Many ground segments currently in development lie somewhere between these two different layered approaches, or they may be a hybrid of the two approaches.

In accordance with current NSS acquisition policy and guidance, most ground segments currently in the acquisition planning stage are expected to be acquired as layered architectures for both the hardware and the software. In these systems, the IS layer is a discrete hardware-software unit that will provide many of the functions listed previously which may be organized into a more cohesive collection/framework of services to support the mission-specific capabilities. The IS layer uses the network layer to support external communications and most internal communications. To support the mission-specific functions, the IS layer provides a set of standardized interfaces that can be accessed by the software applications and services that make up the mission-specific capabilities. The industry is rapidly moving towards organizing these interfaces via standardized protocols, such as SOA employing application programming interfaces (APIs). Increasingly, these common services are organized into a well-defined layer of software called an application service provider (ASP). This is the approach currently envisioned for The Space and Missile Systems Center (SMC) Enterprise Ground Services (EGS) program, which is intended to provide a common ground segment for all future SMC programs, beginning with common services to support satellite telemetry, command, and control (TT&C) functions. The ASP may be considered part of the IS layer, or it may be

considered part of the mission-specific capabilities separate from the IS layer, depending on how the ground segment is being organized and acquired.

NSS ground segments are beginning to embrace current commercial practices successfully used by companies such as Google Inc. and Amazon.com, Inc. Importantly, NSS ground segments are beginning to explore the cloud-based computing approach, in which the processing hardware is organized as a networked set of generic commodity processors running in a data center, and the processing tasks are implemented as virtual machines executing within this generic processing capability. In cloud-based systems, the processing and network capability is provided as a service to support the mission-specific functions above, based on the concept of an infrastructure service provider (ISP). The ISP may bundle together the operating system, virtual machine, processing hardware, and networking layers; and this may blur the distinctions within the ground segment between the IS layer and the mission-specific and network layers.

9.2 Definitions

Application Programming Interface (sometimes application *programmer* interface) (API) A technical approach for defining and publicizing standardized interfaces between software components—frequently but not exclusively associated with the SOA approach.

API Management An approach for implementing and managing inter-process communications in a Service Oriented Architecture (SOA) based on relatively light-weight protocols that broker communications between services by passing RESTful URLs between processes. (See also ESB.)

Application Service Provider (ASP) An increasingly widespread approach for organizing an enterprise in which commonly used functions are provided as services that can be called from mission-specific applications and which are managed by a central authority. Frequently used in conjunction with an infrastructure service provider (ISP).

Cloud computing A technical and organizational approach for providing processing & storage capabilities in which the server hardware is organized as a networked set of generic commodity processors running in a data center, and the processing tasks are implemented as virtual machines executing within this generic processing capability.

Defense in depth A DOD strategy for enhancing the resilience of government systems to cyber-attacks by instituting layered IA protections, rapid reporting mechanisms between peer systems, and greater emphasis on operating through attacks.

Enterprise Ground Services (EGS) A new program at SMC to develop a common ground infrastructure for all future SMC ground segments.

Enterprise Service Bus (ESB) A software construct that may be employed in SOAs to facilitate integration of separately developed software components. The ESB supports inter-process communications, resource discovery, and resource management for load balancing or fault mitigation. (See also API Management.)

Infrastructure Service Provider (ISP) an increasingly widespread approach for acquiring and operating the processing, storage, & networking software/hardware for an enterprise in which the server hardware and some of the lower-level software on each server are centrally managed in a data center and provided to applications as a managed set of infrastructure services.

Representational State Transfer (RESTful) A technical approach for inter-process communications in loosely-coupled systems that underlies the worldwide web and the API management approach to implementing SOAs.

Service Oriented Architecture (SOA) A system/software architectural approach that is based on organizing the components of the system as modular units called services. The services interact with each other to accomplish the mission of the architecture. Services communicate with each other via standardized protocols based on the same principles as the worldwide web. This allows SOAs to scale to large numbers of services executing simultaneously and possibly at geographically distributed locations. There are several different standardized methods for implementing, executing, and managing the protocols that support communications between services and service discovery, such as Enterprise Service Bus (ESB) and API Management.

Uniform Resource Locator (URL) The addressing scheme used in the world wide web and also used to pass information between services in the API management approach for SOAs.

9.3 Detailed Description of the Infrastructure Layer

Ground segments incorporate an IS in one form or another in order to support the following objectives:

- Encourage commonality. (Do in common what is commonly done.) This simplifies the design and maintenance of the system and may reduce software licensing costs.

- Enhance the modularity of the system as a whole. If the interfaces between the IS and the adjacent layers are well-defined and based upon

non-proprietary standards, then the IS and the other portions of the system are insulated from each other. This simplifies the design, implementation, and integration of the ground segment and reduces dependencies between layers. These reduced dependencies in turn facilitate changing one part of the architecture without requiring extensive changes throughout the system.

- Facilitate adoption of an ASP/ISP environment, which may reduce O&M costs. This also enhances the resilience of the architecture by making it easier to migrate the system to a different set of processors or even to a new location if the primary location is disrupted by natural or man-made events.

Key characteristics and usage of the IS includes the functionality assigned to the IS in a given ground segment which varies greatly depending on when it was built or the acquisition lifecycle phase of the project. Generally, what is considered infrastructure can be thought of as a rising tide as more and more higher-level functionality is considered basic infrastructure. The current move towards cloud-based ASP/ISP environments is a timely example of this trend. The following is a representative list of functions that may be assigned to the IS:

- The IS provides a uniform interface for the mission-specific components to access the network layer and other common resources.

- The IS provides a modular interface for external systems to communicate with the ground segment in order to insulate other entities from changes in the ground segment.

- The IS generally provides the capability to monitor and control the ground segment hardware and software. This is often implemented with a commercial off the shelf (COTS) product such as HP OpenView.

- One of the most common functions to be provided by the IS is persistent storage. Different ground segments may provide a range of different storage services. The storage devices are usually centrally managed and backed-up according to retention policies configured by users to meet mission needs. At the low end, the storage service may take the form of merely provisioning processors with extra disk capacity that can be used to perform data intensive processing and storage tasks. At the other end, the IS may provide multiple off-the-shelf database management systems that can be accessed as services (as in SOA) and configured to meet a variety of different mission requirements. The IS may automatically mirror the data to a remote

location to support resilience against both man-made and natural disruptions.

- The IS layer of a newer ground segment is likely to be built using a SOA approach. If the SOA employs an enterprise service bus (ESB) to broker communications between different software components, then the ESB is generally hosted in the IS layer. The ESB enables separately-developed software to be integrated more easily than traditional approaches because the protocols are executed at run time, rather than at compile time.

- The IS provides a platform for development of new and enhanced capabilities to extend the functionality of the ground segment without major modifications to existing hardware and software.

- The IS is one of the most central components of a ground segment with respect to resilience against cyber-attacks. It is one of the most likely portions of the system to be targeted, and is usually responsible for monitoring the interactions of other components to detect anomalous behavior.

Key interfaces, interactions, and transactions with other elements of the architecture include the following:

- The IS layer interfaces to the network layer to access IP-based network connectivity for some of the internal communication between components of the mission-specific capabilities, including communications to remote locations within the same ground segment.

- The IS layer employs the network layer as the interface for the communications external to the ground segment, such as peer systems, remote stakeholders, and the Air Force Satellite Control Network (AFSCN).

- Increasingly, NSS ground segments are employed as elements within a system of systems construct in which the ground segment interacts with peer systems to fulfill missions that cannot be met with individual or stove-piped systems. In addition to the basic network communications provided by the IS, the IS may provide higher-level functions such as support for discovery of resources. These resource discovery mechanisms may employ SOA protocols, even if the ground segment itself is not implemented with a SOA as its internal structure.

- The IS may also incorporate specialized connections, such as: video teleconferencing (VTC); and non-IP connectivity (e.g. time and frequency references or analog signals).

- The IS may provide specialized, high-assurance communications functionality to support urgent messaging interactions to fulfill mission requirements or to support reporting mechanisms for defense in depth as part of cyber-resilience.

- To support the mission-specific functions, the IS layer provides a set of standardized interfaces that can be accessed by the software applications and services that make up the mission-specific capabilities.

9.4 Technical Considerations

Systems employing designs that incorporate a well-defined infrastructure layer benefit from commonality and economies of scale. However, this means that any errors in the IS will potentially impact many components of the ground segment. The lesson learned is to ensure that system-level requirements are systematically flowed to the IS layer, and that the IS layer is subjected to the same degree of scrutiny during design, development, and operations as all other parts of the system. This includes testing for functional requirements as well as system attributes such as reliability, availability, recovery latency, security, etc. Early prototyping of critical infrastructure components and benchmarking of off-the-shelf products can reduce risk in this area.

The IS layer tends to incorporate a very large fraction of COTS components and sometimes open source software. These resources provide significant benefits to customer programs. However, the associated lesson learned is that ground segment developers and maintainers need to exercise great discipline for the off-the-shelf products in the IS considering the following: assessment and selection; maintaining configuration and version control; and planning ahead for timely technology refresh or replacement.

One side effect of moving to an ASP/ISP construct is that the responsibilities for correctness, performance, and mission assurance become more diffused. This is true both throughout the actual ground segment, as well as among the organizations responsible for specifying, acquiring, developing, testing, deploying, operating, and employing the system. There are two significant lessons learned from this concern. The first is to ensure that all stakeholders work closely together to establish trust and also to build robustness into the ground segment to detect, mitigate, and correct problems as early as possible. The second is that clear lines of authority and governance procedures must be

established in tandem with the technical architecture to ensure successful development and effect management.

There have been incidents where modest updates to the infrastructure resulted in serious unintended consequences for a ground segment. The lesson learned from this concern is that the need for thorough testing is just as important during sustainment of the IS as it is during the deployment of the mission-critical applications. The scope of a change is not very predictive of the potential impact. Even modifying a single line of code or changing a single configuration parameter can cripple an entire ground segment at an unpredictable time.

9.5 Programmatic Considerations

The primary programmatic issues to watch for in the IS layer of a ground segment can be broken out by where the project is in the acquisition process. For the IS components of legacy ground segments in operations, the most important considerations are: ensure that the IS components remain reliable; ensure that the IS components are kept up to date with respect to patches and technology refreshes; ensure that sustainment efforts do not disrupt normal operations; focus IS sustainment efforts on the highest priority needs; contain IS sustainment costs; and ensure that IS sustainment efforts do not degrade the reliability, quality, or maintainability of the overall system.

The following are recommended best practices in the O&M phase of a ground segment:

- Identify "what is to follow?" most likely for the system: technology refresh, recapitalization, continued evolution, proliferation, replacement, retirement?

- Assess each proposed sustainment effort to ensure that it is consistent with the current goals of the customer and with what is expected to follow.

- Take advantage of any metrics already provided by the contractor. If the current metrics are inadequate, explore opportunities to strengthen the metrics.

- Explore alternative approaches for sustainment to take advantage of potential opportunities, such as: avoiding unnecessary work, cost savings, reducing dependencies on proprietary hardware and software, and increasing competition.

For IS components of ground segments currently in development, the most important considerations are: ensure that the IS components of the product baseline will support the system requirements; ensure that the most important risks and opportunities for the infrastructure are understood and are reflected in the development plan; ensure that the IS products include sufficient instrumentation to measure progress in development and to operate, trouble-shoot, and maintain the system effectively in operations; ensure that the development plan reflects a realistic approach to test, deploy, and transition the IS products to operations; and ensure that the IS components are sustainable within the constraints of the program.

The following are recommended best practices to consider in the development phase of a ground segment:

- Regularly assess as objectively as possible whether the program is on track according to the baseline and how much margin is available.

- Regularly assess as objectively as possible whether the current program baseline is the right thing to be doing. Are the current requirements and priorities still what the customer really needs?

- Ensure the customer and the contractor efforts are focused on driving down the most important risks.

- Explore opportunities internal and external to the program.

- Take advantage of metrics and other opportunities for insight, and make sure the contractor knows you are watching.

- Maintain awareness of the contractor business case, as well as the customer business case.

For ground segments currently in acquisition planning, the most important considerations are: ensure the overall product architecture reflects an appropriate design that is modular and layered with a clearly defined IS layer; ensure customer needs are clearly expressed in the capability development document, the concept of operations (CONOPs), and other documentation and are flowed-down to the IS components as measurable requirements; ensure the design of the IS layer will support the system requirements; ensure the IS layer supports the quality attributes of the system, such as favoring standards-based interfaces and avoiding unnecessary proprietary features; ensure that the scope and technical approach of the IS product development are feasible within the program's constraints; ensure that the IS product design addresses the most important risks and opportunities; ensure that the IS product design and CONOP realistically

address testing, deployment, transition to operations, sustainment, and retirement.

The following are recommended best practices for the acquisition planning phase of a ground segment:

- Get involved in the development/maturation of the system CONOPs as early as possible, and ensure that users affected by the CONOPs are engaged early as well.

- Identify and assess the key factors that will determine the appropriate acquisition approach for the ground.

- Get the initial government estimate of the scope, effort, schedule, and cost (e.g., cost analysis requirements document) to be as accurate, forthright, and transparent as possible.

- Identify driving requirements, risks, opportunities, and areas of uncertainty affecting ground.

- Ensure the request for proposal (RFP) provides strong support for government data rights.

- Ensure the RFP provides strong support for metrics and appropriate provisions for transparency, oversight, and economic incentives.

The following principles provide high-level guidance about how a modern NSS ground segment can most effectively be designed, implemented, deployed, managed, operated, and evolved throughout the entire ground enterprise lifecycle. The focus of many of these principles is that the ground segment IS layer software components must achieve certain common features implemented with interfaces that are consistent across the architecture. These consistent interfaces for common features will simplify the management of the ground segment as a whole and derive benefits from commonality. However, the semantics, internal implementations, and performance requirements for these common features will vary depending upon the mission context and the application. Most of these design and management principles are applicable mainly to the IS layer of ground segments currently on the drawing board, since they cannot be readily added to an existing product unless they are “baked in” from the early stages of the acquisition. However, some can be applied at any stage of an acquisition. Also, many of these architectural principles can and should be applied to all of the software components of the ground segment and not only the IS.

1. All IS interfaces must be exposed through service interfaces only.
2. All IS interfaces must be published in both machine-readable form (discoverable) and in human-readable form (e.g. a service catalog).
3. All IS interfaces must support any applicable access control, accounting, and auditing features according to common, standardized, non-proprietary interfaces.
4. All IS interfaces must include a basic set of monitor and control features and fault detection, isolation and containment features according to common, standardized, non-proprietary interfaces.
5. All IS interfaces must include a basic set of features to enable recovery or migration according to common, standardized, non-proprietary interfaces, to support continuity of operation.
6. Appropriate metrics and how frequently they are collected and reported must be defined and flowed-down to every IS service interface.
7. New/updated capabilities must be on-boarded into the IS through a well-defined, rigorous, transparent process.
8. Standardize on standards, not products—discourage developers from relying on proprietary features in their implementation of the IS.
9. Incentivize developers, users, and other stakeholders to make use of common services and reuse existing frameworks, services, and applications.
10. Incentivize developers and sustainers to design and implement software to expose reusable functional modules as services and to design software for reuse.

9.6 Bibliography

Lanzinger, Donald J., R.E. Berri, and A.C. Hoheb. *Standard for Interface Development and Support Engineering (IDSE) for the Air Force Satellite Control Network (AFSCN)*. TOR-0091(6488-04)-1 REIS A, The Aerospace Corporation, El Segundo, CA. 1993.

The Object Management Group (OMG) http://www.omg.org/ and other organizations manage technical standards for various elements of information systems that are relevant to NSS ground, such as: metadata tagging, SOAs, and service discovery.

The Network Centric Operations Industry Consortium (NCOIC) www.ncoic.org, of which Aerospace is a member, organizes and promulgates best practices for applying networking standards and SOA best practices to Government programs.

Some Government programs are employing the Joint Architecture Reference Model (JARM) to organize their layered architectures, particularly in an

ASP/ISP environment. http://www.ndia.org/DoDEntArchitecture/Documents/McGovern.pdf

Numerous *de facto* standards exist for cloud-based computing environments, including open standards, commercial products, and open source communities.

An extensive bibliography of resources for ground segment infrastructure with an emphasis on SOA and cloud-based systems can be found in:

Acquisition Guidance for Service-based Information Technology and Software Services: A Literature Survey. TOR-2015-02807, The Aerospace Corporation, El Segundo, CA.

9.7 Acronyms

AFSCN	Air Force Satellite Control Network
API	application programming interfaces
ASP	application service provider
CONOPs	concepts of operations
COTS	commercial off-the-shelf
EGS	Enterprise Ground Services
ESB	enterprise service bus
IA	information assurance
IC	intelligence community
IP	internet protocol
IS	infrastructure service
ISP	infrastructure service provider
JARM	joint architecture reference model
LAN	local area network
NCOIC	Network Centric Operations Industry Consortium
NSS	national security space
O&M	operations and maintenance
OMG	object management group
REST	representational state transfer
RFP	request for proposal
SMC	Space and Missile Systems Center
SOA	service oriented architecture
TT&C	telemetry, command, and control
URL	uniform resource locator
VTC	video teleconferencing
WAN	wide area network

Chapter 10
Systems Engineering

Gail Johnson-Roth
Enterprise Systems Engineering
Corporate Chief Engineer's Office
Robert Sudakow
Systems Integration and Test Office
Mission Assurance Subdivision
Marilee Wheaton
Systems Engineering Division
Engineering and Technology Group

10.1 Introduction

Systems engineering (SE) is the interdisciplinary engineering management process that evolves and verifies an integrated, lifecycle balanced set of system solutions that satisfy customer needs [1]. SE translates a need or identified deficiency into a system architecture through the application of rigorous methods to the iterative process of functional analysis, allocation, implementation, optimization, test, and evaluation [2]. SE incorporates technical parameters to ensure compatibility among physical and functional interfaces, and hardware and software interfaces, in a manner that optimizes system definition and design [3]. It also integrates performance, manufacturing, reliability, maintainability, supportability, global flexibility, scalability, interoperability, upgradability, and other special capabilities into the overall engineering effort [4]. SE focuses on the development and integration of all key elements of a system into an overall system. The combined SE processes are intended to be a comprehensive strategy for the technical development and sustainment of the system. The overall SE process is well defined in numerous technical publications, in general, however the application is critically dependent on the systems engineers with domain expertise to execute relevant to the acquired system [5].

The acquisition program office SE organization includes civilian and military members, system engineering technical assistance (SETA) contractors, and federally funded research and development centers (i.e., The Aerospace Corporation). The program office members are responsible for the technical management in monitoring the prime contractor to ensure delivered products meet requirements. Program offices are usually required to develop a systems engineering plan (SEP) that defines the government technical planning expectations and that should include an overall approach to requirements, technical staffing and organizational planning, technical baseline management, technical review planning, and integration with program management. The SEP

provides guidance for systems engineering as applied to the specific acquisition program and identifies the essential SE activities and required products.

The prime contractor will have their own SE organization responsible for delivering a system to the government that meets the requirements. The contractor is required to deliver a systems engineering management plan (SEMP) that should be consistent with the SEP and defines the contractor's technical planning with details on their processes, tools, and organization. The contractor will include activities from requirements definition to final product delivery with details on each function to include SE products and product characteristics. The SEMP should provide integration of the technical processes and communication across various integrated product teams across the program. The SEMP should also include integration of subcontractor planning [4].

A representative ground segment reference architecture shown in Figure 10-1 illustrates the main subsystems (ground station terminals, mission management, space asset command and control, mission data processing, and distribution) along with the additional functions (training, test, simulation, archive, and retrieval), and the supporting infrastructure services and network layers. Ground segment SE by necessity is a concurrent engineering environment that addresses SE activities, tasks, and products across the system lifecycle, including new development, upgrades, modifications, resolution of deficiencies, and development and/or exploitation of technology.

10.2 Definitions

System An integrated set of products and personnel that interact with one another in an organized or interrelated fashion toward a common purpose that cannot be achieved by any of the products alone or by all of the products without the underlying organization. The integrated products and personnel fulfill manufacturing, verification, integration, deployment, training, operations, support, and disposal functions to provide needed operational capabilities or satisfy objectives [6]. A ground segment may be a segment within a space/ground system or it may be a separate system supporting one or more space systems; For the purpose of this chapter, the term segment will be used.

Segment A major product, service, or facility of the system.

Subsystem An integrated set of assemblies, components, and parts which performs a clearly separated function as part of the segment [7].

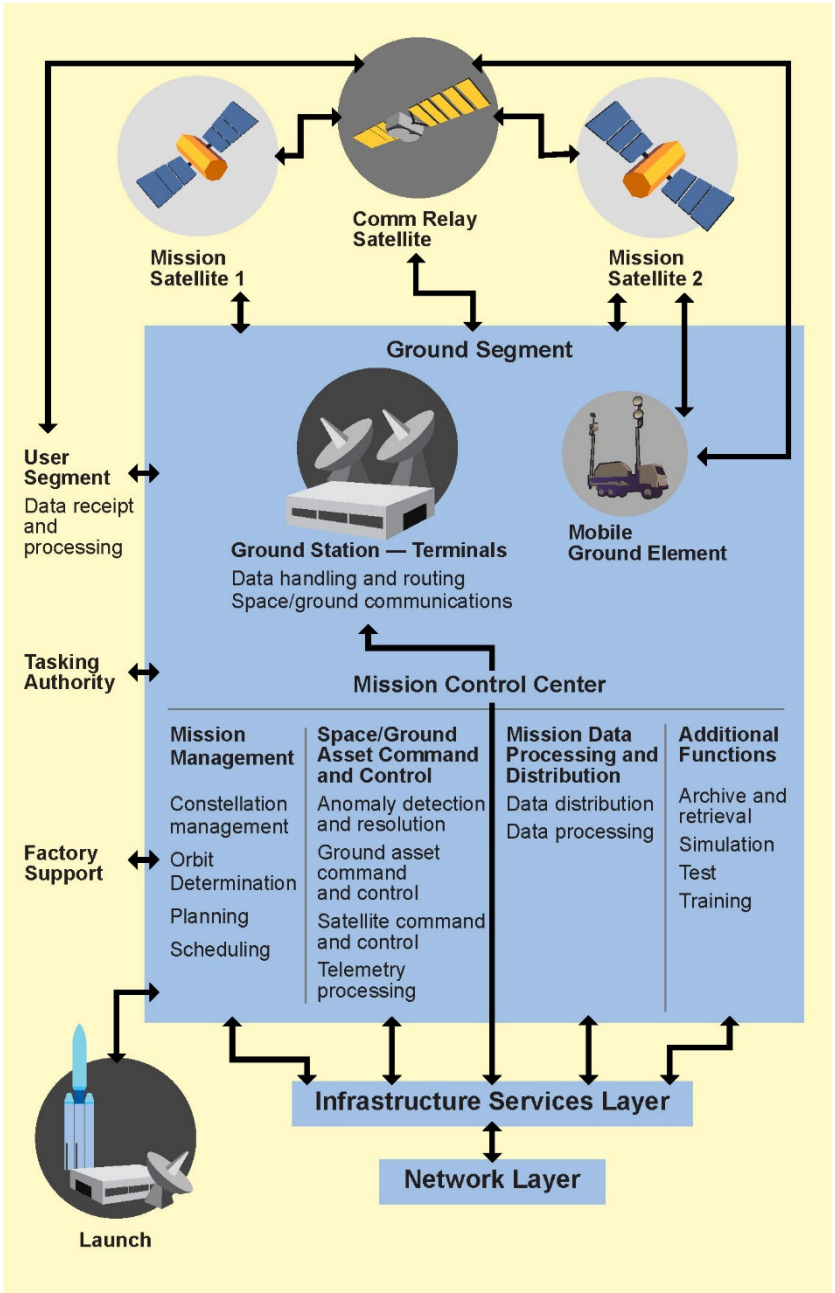


Figure 10-1. Representative ground segment reference architecture with external interfaces.

10.3 Ground Segment Acquisition

Ground segments historically have been acquired as part of the overall ground and/or space system, and also separately to support a specific space system(s). Backup and remote ground station(s) may be added after the initial site is acquired due to political or other site security concerns. Post satellite activities may include multi-mission support composed of elements specific to a mission and elements common to many missions, providing mission data through cloud or other services to existing or new users, and providing an operations center for multiple missions or other needs. In many cases, ground system development activities consist of modification and integration of updates into existing ground sites and systems. Various acquisitions include developments (e.g., builds, spirals, increments) that can be further defined as iterative, incremental, or dynamic baselines. Regardless of the acquisition strategy, all must follow the basic system engineering model.

When a ground segment is acquired as part of a space and/or ground system under the same acquisition lifecycle, some portions of the ground segment may need to be verified prior to launch of the space vehicle. Portions of the ground segment are usually verified through the use of a verified space segment simulator which is acquired as a component of the ground segment. Consequently portions of the ground segment may be in different phases of the lifecycle at the same time. In some cases, the full ground segment capability is required after the satellite completes on-orbit test, so the ground segment must pass ground system-level and readiness testing before launch. After the ground system is operational, additional satellites may be launched with or without significant changes. In these cases, ground segment test before launch is tailored to ensure the new satellite operates properly with the current or upgraded ground segment. A large challenge is to deliver new capabilities, or even an entire new segment, into an operational facility without impacting current operations.

When a ground segment is acquired separately for the space segment, the lifecycle generally follows the lifecycle of the associated space segment(s) since requirements are dependent on the space segment design and performance. The ground segment may be developed incrementally to support space system needs, and may therefore be in different phases of the lifecycle simultaneously.

Ground segments may be acquired to support multiple space systems with common infrastructure, and both common and unique program support functions. In this case, the ground segment acquisition lifecycle may be independent of the associated space segments, but with planned increments that can be phased into the space system lifecycles as appropriate. Ground segment pre-launch testing will ensure that the ground segment can support the required mix of satellites. A common ground segment acquisition strategy is a major upgrade to an existing ground system. In this case, the lifecycle would be

independent of the space segment but coordinated with any space segment updates or schedule needs.

Defining and documenting the ground segment, sub-systems, and hardware and/or software configuration items involves knowledge of the detailed requirements engineering discipline that consists of requirements development and requirements management executed by the contractor with oversight by the acquisition program office. Verification must target a strategy acceptable to the acquisition authority that eventually includes an integrated ground system hardware and software that ensures design adequacy and correct implementation prior to being certified as being operationally ready.

Figure 10-2 illustrates the standard SE diagram—the left side of the SE "Vee" depicts the contractual requirements illustrating the decomposition process down to the design and manufacture and/or build requirements, while the right side shows the companion verification and validation at each level of requirements [6]. Regardless of the acquisition type, basic SE core processes apply to confirm the specification requirements are satisfied in all configurations and conditions encountered for the mission.

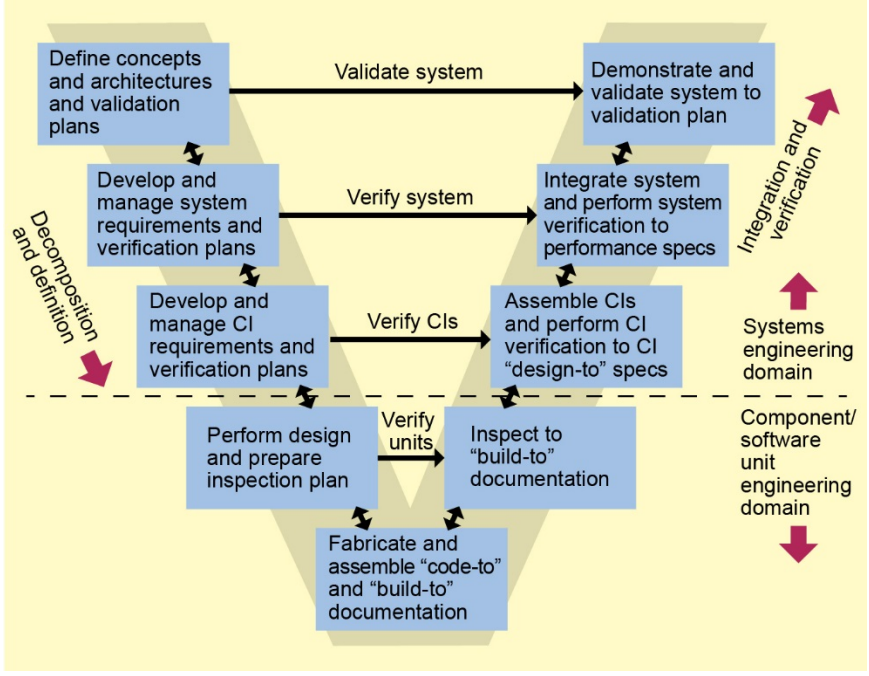


Figure 10-2. Systems engineering diagram [6].

10.4 Core System Engineering Processes

The SE process can be partitioned into two interdependent parts: the technical management process and the technical process. The technical management process uses the program management plans and stakeholder information to create the program technical plans. The technical management must tie in closely with the program management function. Additionally, the program manager must work closely with the contractor to develop and explore solutions to arrive at an optimal systems solution that meets cost, schedule, performance and risk goals. As with any large complex system, schedule pressure and completion of planned development to include integration and test activities is nearly always a challenge due to resource constraints and competing activities. Key is to ensure early engagement and accurate definition, as re-planning almost always comes with increased risk and higher cost.

The technical plans are used to control and assess the technical process. The technical process establishes the technical baseline for the system requirements, the designs, and the delivered products, and then verifies that program products meet the established baseline, allocated cost, schedule, maturity, and performance. The technical baseline is applied to the design of the system and the enabling products to develop, produce, deploy, sustain, and service the program end products. There are a number of defined core SE processes with supporting disciplines that serve as the framework for mission assurance oversight by the acquisition authority that is largely executed by the SE organization. Figure 10-3, derived from the *Mission Assurance Guide*, illustrates this concept [8].

There are a number independent reviews, gates, audits, and assessments required and/or executed by the acquisition authority throughout the lifecycle of the program to assess the technical maturity, evaluate risks and opportunities, understand stakeholder expectations, and ensure readiness for the next phase of the program. As part of any rigorous SE process, technical reviews are conducted at logical points in the program or at key milestones. Key milestones are defined by the specific acquisition authority with expected deliverables and satisfaction of exit criteria [9]. Standard milestones include concept refinements, technical development, system development and demonstration, production and deployment, and operations. Each type of audit varies by purpose, scope, depth, and time phasing. As the acquisition program moves through the lifecycle, the reviews and audits become more detailed and definitive with specified requirements for the technical reviews and audits [10].

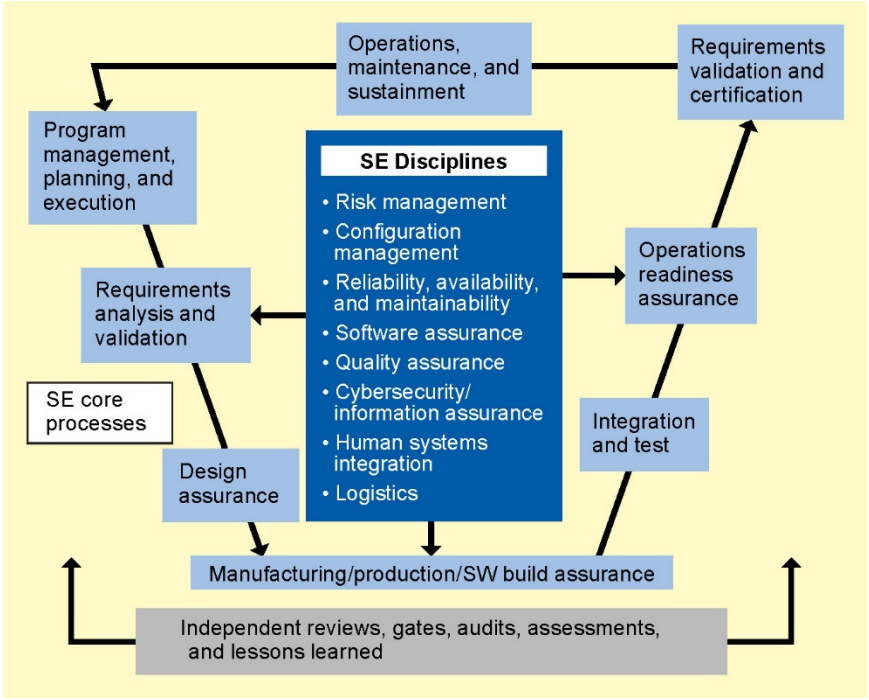


Figure 10-3. Core system engineering processes and disciplines.

10.4.1 Requirements Analysis and Validation

Requirements analysis and validation is a critical SE core process for all systems. Requirements development is the SE process of taking input from relevant stakeholders and translating that input into technical requirements by conducting requirements analysis, functional analysis, and allocation and requirements synthesis [8]. The program office can provide the initial requirements set for ground segment acquisition. While the contractor is responsible for mission analysis, requirements and specification development, verification planning, and system validation activities, the government program office team performs associated independent technical assessment requirements validation activities. Technical baseline management requires identification and resolution of requirements issues in coordination with the contractor and the user.

Requirements synthesis defines and designs solutions for each logical set of functional and performance requirements within the functional architecture and integrates them as a physical architecture. Outputs of the synthesis include: determination of completeness of the functional and performance requirements for the design, definition of internal and external physical interfaces,

identification of critical parameters, defined system and segment solutions to enable verification; and translation of the architecture into a work breakdown structure, specification tree and configuration baselines [8]. Parallel government and contractor processes involve a set of orderly tasks using analytical tools and simulations to synthesize, develop, and ensure a consistent set of program requirements that meet user needs within affordable costs and acceptable schedules. User needs and mission requirements are optimized and decomposed into system requirements and flowed to build-to specifications and interfaces. Requirements validation provides confidence that the technical means and processes meet user needs. Requirements validation occurs during the front end of the systems acquisition life. The government typically performs an independent analysis of the contractor's models and simulations used to assert the performance represented by the system requirement set.

When a ground segment is acquired as part of a space and/or ground system, the ground segment requirements are flowed down from and traced to a system-level specification. The ground segment specification will contain derived requirements based on architecture and environment. Independently acquired ground segments will have a system-level specification with references to key space segment specifications and other external drivers. Heritage components of the ground system typically have heritage requirements that may need to be updated for the new ground segment. When commercial, off-the-shelf (COTS) components are used for significant ground system functions, the requirements may have to be tailored to meet the as-built COTS product.

Multiple sources drive ground segment requirements. Operational requirements drive basic technology and architecture of both space and ground. Operational requirements are defined in the concepts of operations (CONOPS). The CONOPS methodology is developed early in the program and provides an allocation of what will be done on the ground as it relates to the rest of the system-defining requirements in each area. Contractors usually develop an operations concept derived from the customer provided CONOPS to sufficiently describe the system architecture and operational modes to help provide traceability between various design specifications and operating procedures.

Space segment capabilities (orbital dynamics, payload sensor characteristics, data transmission bandwidth, etc.) drive requirements for ground segment architecture, communication capability, data handling, and processing capability. User needs also drive ground segment requirements for external communications, mission data processing, security, reporting, etc. Ground segments will have various interface control documents (ICDs). The most common are space-to-ground ICDs controlling telemetry and commanding of the space vehicle, and external user documents controlling reports and other outputs from the ground segment. The ICDs are also defined between

cooperating ground segments and within the elements of the ground segment being developed.

Since ground segments will have an operator interface, the human-to-computer interface is a critical aspect of the segment requirements. These requirements will depend on the expected skill level of the operators (ranging from entry-level military officers to experienced contractor personnel) and the complexity of the ground operations (ranging from basic satellite communications to complex mission data analysis). The need for trained operators also drives requirements for training systems. The trend today for training systems is to build stand-alone, computer-based training systems that share common infrastructure. However, in some cases the training may involve support from high-fidelity simulations to train operators on satellite anomaly resolution, orbital and attitude maneuvers, appendage deployment, on-board computer support, and other complex operator functions.

10.4.2 Design Assurance

The design assurance process is an iterative set of planning, analysis, test, and inspection activities performed from conceptual to preliminary detailed design stages to improve the probability that the ground segment meets the intended requirements through all operating conditions and throughout the design life. The design assurance activities include the assessment of the design and evaluation of the product qualification, manufacturing/build, and test phases. Design assurance includes: mission design, system and/or segment design, and hardware and software design and test. Software and hardware metrics can be used to support the design assurance process and to monitor attributes such as quality and test satisfaction [8].

Design assurance ensures compliance with detailed design requirements as reflected in appropriate specifications and standards through the product qualification, manufacturing, and test phases. Assessment of the readiness to manufacture hardware and build software includes ensuring the hardware can be manufactured and the software can be built to meet performance, cost, and schedule requirements. Employing in-depth design audits with rigorous in-depth processes and with the appropriate subject matter expertise is essential to ensure the selected design is the right design. Design reviews should include detailed critique of the design (documentation, drawings, analysis, and test data) by independent reviewers relative to formal requirements. The detailed review should be independent and separate from the program milestone, critical design review event [11].

10.4.3 Manufacturing/Build Assurance

"Manufacturing engineering encompasses the use of available and certified materials, parts, and manufacturing processes to create products that fulfill document design requirements" [8]. Manufacturing engineers work closely with SE, as well as design engineering, parts, materials, processes engineering, test engineering, reliability engineering, safety engineering, quality assurance, and configuration management to ensure production of high-quality products that meet the as-designed requirements. The contractor is required to convert designs into manufactured products, processes, and methods of production and products. The government program office must ensure the manufacturing processes are able to produce hardware and software that meets the design requirements.

The ground segment is dependent on complex software for successful operation and mission execution. Ground software supports routine and anomalous space vehicle operations, and also performs mission planning, mission data processing, and mission data dissemination. The government is often required to create the software acquisition management plan, which is leveraged to identify segment risks and feeds into the request for proposal. SE plays a key role in communicating the program as a whole to include the software acquisition and development environment within the program. There are several software CDRLs that identify all data items to be delivered, one of the key documents is the software development plan (SDP). The government team conducts technical reviews of the work products of the suppliers developing the software to include plans, procedures, processes, products, measurement data, and activities to determine technical accuracy, completeness, and quality, and to identify any shortfalls that may impact mission success [8].

10.4.4 Integration and Verification

Requirements verification is a SE activity that requires developed plans and defined actions to prove the as-built item complies with the requirements baseline as determined by test, analysis, demonstration, or inspection methods. The requirements for functional, performance, and environmental testing of a ground segment is detailed in Aerospace TOR-2013-00175, and are applicable to the design, development and integration of new, reused, or modified software and hardware in a ground segment [12]. The requirements may also be applied to simulators used for training operations and maintainers of a ground segment.

The verification activity is performed from the lowest-level configuration item to the system-level, as shown in Figure 10-2. The selected verification methods are applied at the appropriate and lowest level of assembly where the selected method is most perceptive at providing the needed verification. Requirements at the configuration item level and the system or segment level are verified within the SE domain.

System-level validation occurs before the as-built system is transitioned into mission operation to validate the correct segment was built. Performance validation consists of various types of test and demonstrations such as day-in-life test, test-like-you-fly (TLYF) process, soak testing, and scenario tests representing normal and stressing operations. TLYF is a prelaunch/pre-operational SE process that translates mission operations concepts into perceptive operability tests to assess the risk of any potential missing mission-critical areas where it is not feasible to test or adequately represent key mission characteristics while executing a test that is documented as an exception to the process [13].

Ground segment integration includes integration of the operational hardware, software, and data in the ground segment, including computers, archives, files, databases, workstations, telemetry and command processing hardware, network components, and communications hardware. Software integration starts with software units with their associated data, and builds up to components, subsystems, and the full software delivery. This effort usually is accomplished in a contractor's development and test environment. The final stage of ground segment integration is the integration of the software and data with the operational hardware in the operational environment. The entire integration process is highly controlled to ensure that the segment is ready for formal verification.

Verification and validation testing may rely on test environments and simulators for portions of the operational system that are not available for this testing. The most significant interface required for the ground segment is the interface of the space segment with the flight computers and onboard software. This interface is usually supplied by a software simulator that provides inputs and responses to the ground segment based on the space-to-ground ICD. In some cases, a test assembly consisting of some or all of the actual space hardware, including the onboard computers and flight software, can be used for interface testing of the ground segment. When available, this type of test assembly is the most effective means of testing space-to-ground interfaces.

The contractor SE integration team (SEIT) is responsible for generating the program-integrated test and evaluation (T&E) plan, providing objectives and guidance for the generation of individual system test plans (STPs), verifying effectiveness and suitability. Effectiveness is an objective execution demonstration to ensure the system works as required and is effective; suitability includes operational soak to ensure the system is stable and dependable.

The STPs are developed, reviewed, approved, and released through the program configuration control board process. STPs are developed for each major incremental, delivered capability and include associated support requirements for the segment operational test and evaluation (OT&E) directed by the

government. In addition, STP baselines provide the detailed test requirements for a given training event. Planning incorporates schedule spans for procedure dry-runs, and the provision for formally documenting anomalies and failures via a discrepancy report process.

The integrated T&E plan, and individual test plans, are reviewed and approved via the change board process. Each training event will be baselined in the program integrated management schedule (IMS) and linked with predecessor ground segment software deliveries, when appropriate.

While the requirements verification plan defines T&E requirements, integrated product teams (IPTs) derive detailed test requirements and develop plans and procedures for integrating, testing, and verifying the products throughout the program lifecycle. Each IPT provides a common test team for factory, launch support, and early orbit test (EOT). The SEIT integrates the activities by approving and directing intersegment tests. Segment test readiness events utilize the same objectives conducted in integration, test, and evaluation for formal verification in order to ensure the system-like test reduces risk. The system test and segment test flow of information supports combined development/ operational test planning and test opportunities supporting transition from developmental test and evaluation (DT&E) to OT&E, and mission and interoperability certification. The acquisition authority conducts independent technical review of verification plans and products across all functional domains and provides technical guidance and oversight to interface activities as well.

The government program office SE team provides the philosophy on how to test system-level effectiveness and suitability (i.e., how the ground segment tests in relation to the overall system that the segment will support), how the system-level test program evolves from the lower-level segment test programs, and how to support the certification of segment readiness for dedicated OT&E. This SE team also summarize the planning and integration support processes for verification, modeling and simulation activities, and support OT&E and segment certification. Government program office personnel typically observe and witness test and verification events, which requires a strategy consistent with the contractor's development to ensure appropriate oversight and to provide insight into test and verification events. In preparation for the test events, the test documents, plans, and procedures are reviewed, and the TLYF exceptions are independently assessed.

10.4.5 Certification

Certifications are a formal communication to the approval authority that the segment meets specific requirements. The certifications drive SE planning and should be factored into the technical planning to ensure processes include applicable scheduled time and resources. The SEP should provide a certification

matrix with identified, applicable, technical certifications required during the acquisition lifecycle. These events should then be transposed to the contractor's IMS and integrated management plan (IMP).

Certifications may include, but not limited to, spectrum certification compliance, OT&E readiness, radio frequency hazards certification, registration of mission-critical and mission essential information systems, independent logistics assessment, system threat assessment, and OT&E readiness. Many of these certifications entail a lengthy process. As an example, the OT&E events progress in a building-block manner beginning with analyses, modeling, validation of simulation, and ending with field tests. Modeling, simulation, and test beds are used to assess areas in which field-testing cannot be conducted. The program office is required to provide approach guidance to satisfy compliance.

Certification includes a sell-off of effectiveness that concludes at the end of the contractor's test series. At that time, the contractor proves through requirements verification that the element or segment or system requirements and ICDs are met as appropriate. In addition, the program office and/or Air Force Operational Test and Evaluation Center (or like organization) projects compliance against operation requirement documents to predict if the system can pass the subsequent operational test events.

10.4.6 Transition, Operations, and Sustainment

A space system program often includes multiple program activities that are conducted concurrently to include current operations, sustainment of operations (defect corrections and modifications), segment incremental development and production activities, incremental deployment activities, and incremental transition activities that involve multiple agencies associated with test, certification, and operations. Concurrent activities require a balanced approach to allocating available program resources that presents significant complexity for program execution. Transition and operations planning must consider all stakeholders to ensure an effective deployment and transition so as not to significantly alter the segment technical baseline in terms of specifications or design. Similarly, changes arising from implementation of new requirements may impact the technical baseline to affect transition and operations and should be considered when testing the new segment and/or capability.

Once the ground segment is developed, it generally continues to evolve and improve as the associated space systems are replenished and improved. Over time, the technology underlying the hardware and software of the ground segment becomes obsolete or unsupportable. For these enduring ground segments a technology refresh effort must be accomplished to update the hardware and modernize the software. Legacy segment sustainment, maintaining

training, and updating a concept for operational spares are challenges for complex ground segments.

As the ground segment becomes operational, some amount of latent defects surface and must be corrected. Also, some new requirements are usually added to cover emerging system needs and resolution of operational issues. These changes are usually accomplished through a long-term maintenance contract. Typically, this type of contract goes to the development contractor, but with adequate documentation, it is possible to complete this effort with different contractors. These contracts follow the same core system engineering processes and supporting discipline processes to ensure ground segment mission success through operations, maintenance, and sustainment.

The space and/or air systems that are supported by the ground segment also encounter operational problems or updates that must be accommodated. These updates may require a new contract or may be covered by a maintenance contract. The factory- and site-sustained operations require many complex simulations, tools, COTS programs, and software programs, in addition to custom hardware. These require the same maintenance, periodic COTS updates, hardware replacement, and upgrades to improve operations to accommodate changes to the space segment baseline and changes in the missions.

10.5 Key Documentation

10.5.1 Systems Engineering Plan

The SEP is a government document that defines SE aspects on a particular program that details the execution, management, and control of the technical program aspects from conception to disposal. There are typically five critical focus areas: program requirements, technical staffing and organization planning, technical baseline management, technical review planning, and integration with overall management of the program.

There are a number of key documents that should be referenced to ensure agreement with the program's technical development and evaluation approach. The following government generated documents should be referenced in the SEP:

- Initial Capabilities Document
- Capability Development Document (CDD)
- Capability Production Document (CPD)
- CONOPS
- Acquisition Strategy
- Risk Management Plan

- Test and Evaluation Master Plan (TEMP)

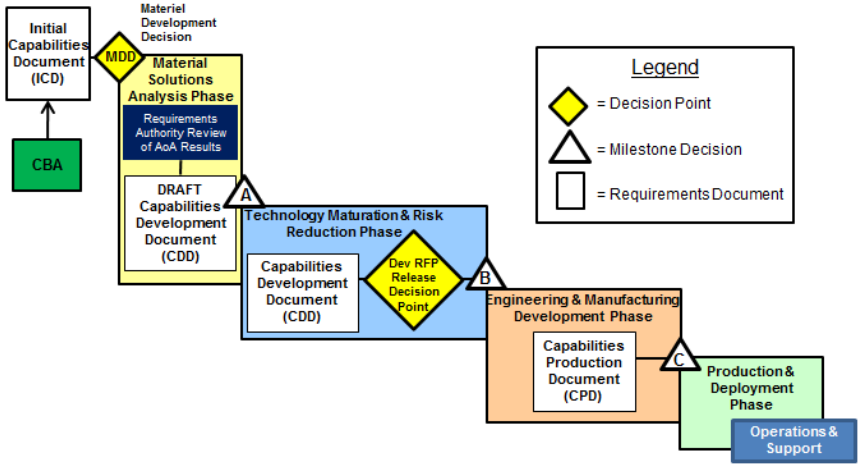
The SEP serves as a stakeholder reference and provides the program's plan to accommodate cost, schedule, and performance in consideration of sustainment trades [14].

10.5.2 Acquisition Documentation

The acquisition authority develops a number of documents as the initial capabilities are defined in the context of an operational concept to meet user needs. Figure 10-4 illustrates the DOD approach to mature the documentation to capture and refine requirements throughout the lifecycle. Other acquisition agencies follow similar processes, and although the actual document titles may differ, the intent and acquisition authority accountability is similar.

The initial capabilities documents satisfy the government's need for a materiel approach that identifies a specific capability gap. The gap is derived from an initial analysis of material approaches executed by the user. The initial capabilities document includes descriptions of the gap in terms of functional area, relevant range of military operations, and the desired effects and time. An outcome is a CDD-generated prior to the request for proposal during the technology maturation and risk reduction of initial program formulation (pre-milestone A). The CDD specifies the operational requirements for the system. As the program matures (post milestone B), the CPD capture the information necessary to support production, testing, and deployment within the acquisition strategy. The CONOPS documents the current and new and/or proposed capabilities required and includes how the system will be used from viewpoints of various stakeholders. This document bridges the initial capabilities to specific technical requirements needed and is used to write and refine the CCD [15].

The acquisition strategy document is a comprehensive plan that identifies and describes the acquisition approach to manage program risk and meet program objectives. This document guides program execution across the program lifecycle and is required in DOD systems to be updated at every major milestone review. Technical data needs for the system is one required component of this document. The acquisition strategy should also summarize the risk management process for the program [16].



CBA = capabilities based assessments, MDD = material development document, AoA = analysis of alternatives, and RFP = request for proposal.

Figure 10-4. Acquisition documentation generated by the government over the system lifecycle [16].

10.5.3 Test and Evaluation Master Plan

The TEMP describes the planed T&E over the program's lifecycle. The TEMP should include applicable test scenarios, appropriate data collection, and performance evaluation criteria. T&E should be integrated with the overall acquisition strategy and drives the generation of detailed T&E plans and resources. Significant events should be defined in the contractor's IMS. The TEMP should serve as a guide for specific test events and integration of detailed test plans to include developmental and operation T&E activities. Defined technical performance measures (TPMs) should be verifiable, and include both nominal and off-nominal conditions. Details of the contractor's development, integration, and certification strategy should be included for each acquisition phase to include any test limitations that may impact results related to critical technical parameters. The overall goals and objectives of the developmental test and evaluation should include specific exit criteria, system acceptance tests, qualification tests, and test readiness reviews [8].

10.5.4 Systems Engineering and Contractor Deliverables

In the development of national security space systems, including the ground segment, it is the responsibility of the acquisition authority (government) to ensure that adequate SE process are implemented. SE data requirements are established by the contract date requirements list (CDRL). These CDRL items

enable the government to specify what SE data is required and when it is needed [6].

SE CDRL items are:

- Systems Engineering Management Plan
- Risk Management Plan
- Integrated Management Schedule
- Contract Work Breakdown Structure
- Software Development Plan
- Configuration Management Plan
- System Verification Deliverables
- System/Segment Interface Control Specification
- Logistics Management Information
- System/Subsystem Specification
- Design Review Information Package
- Engineering Change Proposal
- Failure Summary and Analysis
- Data Accession List (DAL)
- Human Systems Integration Plan

10.5.4.1 Contractor Systems Engineering Management

The Systems Engineering Management Plan (SEMP) is a contracted deliverable that includes the contractor's proposed processes and activities for planning, controlling, and conducting an integrated systems engineering effort. Content should include insight into application of contractor's standards, capability models, configuration management, and SE tools used by their organization. A SEMP should address the following [17]:

- Organization along with location and facility information
- Technical management
- Evaluation and decision making processes
- Risk management
- SE methodology
  - Configuration management process
  - Requirements verification and validation descriptions to include use cases
  - Architecture and design process
  - Software development
  - Hardware development
  - Build management process
  - Testing process
- External interface development and management

- Data conversion development and management
- Implementation planning (i.e., system deployment, training)
- Production support strategy

10.5.4.2 Risk Management

Risk management is the identification, assessment, and prioritization of risks (the effect of uncertainty on objectives) followed by coordinated and economical application of resources to minimize, monitor, and control the probability of and/or impact of unfortunate events. Risks are managed throughout the acquisition lifecycle at the system and segment levels. System-level risks are usually managed through a government-run risk board. Ground-segment-level risks may be managed by a government board or by a contractor board with government cognizance.

Some common ground-level-risk areas are key requirements with tight performance or accuracy requirements, space segment hardware or flight software changes that significantly impact the ground system (i.e., ground system's architecture, complexity, timing, and accuracy), dependence on external data or equipment, and uncertainty in scope and/or complexity of the software development effort, uncertainty of schedule and cost due to programmatic constraints.

The acquisition program office should develop and maintain a separate risk management process to allow for comprehensive assessment and handling of program risks. Program decisions that include additional risk acceptance should be documented in terms of impacts to the technical baseline, enabling understanding of impacts to relevant SE processes and products

The risk management policy should be stipulated by contract with a deliverable that includes a risk management plan that contains the risk index scheme to define the magnitude of risk in terms of severity and likelihood. The risk management process should include SE policies, practices, and procedures. Risk mitigation strategies should be identified and documented to include accepted risk and planned risk reduction (including technology maturity and technology off ramps). The risk process applies to subcontractors and critical vendors who should also include SE processes [8].

10.5.4.3 Schedule Documentation

The integrated master plan (IMP) is a contractual description of the events, significant accomplishments, significant accomplishment criteria, applicable documents, and critical processes necessary to satisfy all contract requirements. The IMP defines the SE processes and compliance documents to be used by the

prime contractors and the subordinate contractors. The completion of each significant accomplishment is determined by measurable, significant accomplishment criteria. The significant accomplishments have a logical relationship to each other (such as parallel or sequential) and, in subsets, prepare for events. Narratives that include objectives, governing documentation, and an approach describe the critical processes. The IMP includes an indexing scheme that links each significant accomplishment to the associated contract work breakdown structure (CWBS) element, event, significant accomplishment criteria, and tasks presented in the integrated master schedule (IMS). The data in the IMP defines the necessary accomplishments for each event, both for the contract as a whole and for each team or manager responsible for a specific contractor work breakdown structure (WBS) element [6].

The WBS is a system, product-oriented hierarchical tree composed of the hardware to be developed, produced, or sustained; software to be developed or sustained; services (including cross-product activities such as systems engineering); data; and facilities that encompass all work to be carried out under the program or contract, along with a dictionary of the entries in the tree. The CWBS is prepared in accordance with the contract [6].

The IMS shows a time-based plan with detailed tasks. The schedule shows the time relationship between events, significant accomplishments, and the detailed tasks required to complete the contract including calendar dates, time spans, critical path, and slack. The IMS applies the same indexing as used in the IMP. IMS tasks are directly traceable to plans and accomplishments of the Earned Value Management System (EVMS), though some EVMS plans or work packages, such as those for level-of-effort tasks, need not be traceable to the IMS. The IMP, IMS, and the WBS are key tools that should be referenced by the SEP, and used by the program office to ensure overall integration with overall management of the program for oversight of the work and product delivery.

10.5.4.4 Software Development

Ground segments are software-intensive systems and require a software development plan (SDP) as a contract deliverable. The term software development includes: new development, modification, reuse, reengineering, maintenance, and other activities resulting in software products. The plan provides program-unique tailoring for the development effort at hand. The acquirer is provided insight into, and a tool for monitoring, the processes to be followed for software development. The SDP includes details as to the methods to be used; the approach to be followed for each activity; and project schedules, organization, and resources. The SDP focuses on methods and approaches to software production, e.g., testing strategies and programming methodologies. The SDP references specific standards, methods, software and systems engineering tasks, tools, actions, reuse strategy, and responsibility associated

with the development and qualification of all requirements, including safety and security. The SEP should reference the SDP, and integrate the software planning into the overall SE planning. This ensures proper allocation of resources, schedule development, and overall understanding of software development, in terms of critical path events and interfaces [18].

Software on a complex ground segment can take an incremental or an iterative development approach. The incremental approach includes creating a staging and scheduling strategy in which parts of the segment is developed at different times, and integrated as they are completed. The work is broken in sequential iterations, each composed with sufficient analysis, design, implementation, and testing to produce a partially complete system that is stable, integrated, tested, and releasable to the customer. An iterative approach includes a rework scheduling strategy in which time is set aside to revise and improve parts of the system in iterations. Iterations may not by themselves have sufficient analysis, design, implementation and testing to produce a stable, integrated, and tested product that is releasable to the customer [19]. Different approaches may lead to different outcomes if not appropriately managed to ensure SE efforts are not reduced. Schedule volatility, test durations, and discrepancy report detections will vary based on the iterative development approach.

Software metrics measure properties of software, and the development process, (i.e., size, effort, schedule, and quality) [20]. The goal is obtaining objective, reproducible, and quantifiable measurements used in activities such as schedule and budget planning, cost estimation, quality assurance testing, software debugging, software performance optimization, and optimal personnel task assignments. Key to understanding the metrics is recognizing the appropriate and applicable metrics as aligned to the development approach and ensuring that a robust SE approach is inserted to avoid excessive cost and schedule impacts.

10.5.4.5 Configuration Management and Data Management

Configuration management is the process for establishing and maintaining consistency of a product's performance, functional and physical attributes with defined requirements, design, and operation throughout its life. Configuration control boards (CCBs) are established to review and approve changes to controlled documents. Top-level program documents, including the system-level specification, are usually controlled by a government-run CCB, while lower-level documents are usually controlled by a contractor-run CCB with government cognizance. The board(s) approve or disapprove proposed engineering changes and deviations to CIs and baseline-configuration documentation. The contractor is required to deliver a configuration management plan [21].

Contractors typically use a traceability tool, e.g. DOORS, to manage the configuration and traceability of requirements. The scope of the ground hardware and software that must be controlled includes all operational hardware and software, as well as all factory equipment and other test equipment used for system test and verification. There are various configuration control tools, e.g. ClearCase, to help manage the configuration of the software.

Data management (DM) is a companion function to CM. Much of the data produced on a program is technical in nature and describes a technical result, a plan to achieve the result, and/or the basis for the result. While specifications, hardware drawings, and software code are usually covered by CM, other documentation such as program directives, program procedures, design-to-cost reports, program schedules, risk analysis reports, etc., have a major role in controlling program costs and should be controlled data. Hence, the content, control, and archiving of the data must be managed as a part of the SE process. The program office needs to implement formal DM processes within the government team.

10.5.4.6 System Verification Deliverables

The segment design and development is verified to ensure satisfaction of all document requirements. The SE CDRL list includes system verification deliverables. The contractor is required to provide a documented system verification program, verification results, design qualification data, and acceptance verification data. The product configuration baseline is validated, approved, and maintained throughout the verification process. Segment products to include new, modified, or COTS are verified in the allocated baseline. The program should include system effectiveness evaluation and manufacturing process proofing, and address the requirements and procedures need to verify critical verification methods and processes. Discrepancies with respect to the product configuration baseline, technical performance metrics, and constraints are tracked and reported. The product configuration baseline is formed after confirmation that the as-built, as-procured, as-integrated product is verified for delivery [6].

10.5.4.7 Interface Control

Interface control planning defines the methods used to develop and control internal and external interfaces. All formal interfaces are documented in ICDs. The ICDs are derived from existing plans and further developed as required by the evolving architecture or new requirements. The contractor is usually responsible for developing internal ICD and processing the documents through their configuration boards to include changes to those documents. All external interfaces should be defined with appropriate organizations and agencies

identified with established points of contract. The acquisition program office is the ICD approval authority for documents related to external interfaces

The document that defines the interfaces between the space segment and the ground segment covers all space assets and the interface to the various ground entities including the mission control segment, any relay ground stations, and any specifics to the mission per the system architecture. The interface documentation is updated, via the configuration control process.

The adequacy of the SE processes both on the contractor and government organizations should be assessed to ensure the interface control process is well defined and managed at the program level, with acquisition authority leadership as part of the process to resolve and conflicts between associated contractors or other stakeholders. Interface control working groups should include the acquisition authority SE organization as members. Systems engineering involvement in interface development is critical, particularly when synchronization is required between legacy and development segments, to avoid implementation and verification during transition of functions. Interface changes require clear documentation and full stakeholder coordination [22].

10.5.4.8 Cybersecurity and Information Assurance

Cybersecurity focuses on preventing and defending against malicious attacks and unauthorized use of computer systems. Information security focuses on protecting digital and non-digital information assets to ensure that information systems will perform as needed when needed and be accessible for authorized users only.

For national security space systems, a system security plan generated by the acquisition authority categorizes the system confidentiality, integrity, and availability impact. The plan should include a description of the information system to include the security perimeter and enterprise architecture, and to provide an overview of security controls and requirements and mechanisms for meeting those controls. A separate cybersecurity plan should be created that identifies system-specific security controls. Much of this information is capitulated further in a program protection plan (also generated by the government organization) that includes threat information [8]. Although the contractor may not have a specific contract deliverable, there will be specific requirements that will be levied for cybersecurity and information assurance on the contract. Much of the information crafted in the government plans specific to a ground segment or specific program is sensitive and/or classified, and may be isolated for consideration by the SE organization from the system perspective.

Given some recent issues in some mission operations, it has become clear that the current processes employed to determine mission readiness or the risk

posture of a system do not always sufficiently address the effects cybersecurity mechanisms may have on a mission. Traditional failure mode effects and criticality analysis does not currently take fully into account cyber operations or cybersecurity. Additionally, the current paradigm for addressing cybersecurity for our national security systems is to perform certification and accreditation activities which are geared towards compliance with controls [23, 24] and scanning the system for known vulnerabilities. The ability to successfully carry out the mission is rarely, if ever, taken into account when assessing the security posture of the system [25].

The SE organization should support the applicable information security activities and working groups, as well as review certifications and accreditation packages to ensure completeness and potential risks from a system perspective. Additionally, there will be a coordination role for unique cybersecurity testing associated with network devices (computers, routers, switches, ports, protocols, and services) and associated discrepancy reports that likely will require independent assessment of findings, as well as additional assessments after the contractor implements mitigations. Independent testing and assessment can provide useful insight into cybersecurity posture and compliance.

10.5.4.9 Human Systems Integration

Human systems integration (HSI) is a major component of the ground segment integration. The HSI plan is a key contract deliverable to be reviewed and monitored by the SE organization. Human activity considered by HSI includes operating, maintaining, and supporting the system. HSI ensures workflows and interfaces between operators and machines are optimized for efficient operations and are designed to reduce operator errors. HSI also considers training and training services, as well as the infrastructure used for operations and support. HSI incorporates the following domains as integration considerations: manpower, personnel, training, human factors engineering, occupational health, environment, safety, habitability, and human survivability.

10.6 Technical Considerations

10.6.1 Trade Space

Mission requirements are evaluated in terms of how much capability should be fulfilled by a particular segment—space or ground. The trade-offs consider cost, schedule, technology readiness, security, and even political considerations. Example trades for system architect decision-makers' consideration include: should data be processed on orbit or on the ground, to use satellite crosslinks or build more tracking stations, and whether to increase satellite transmitter power or use larger terminals [26]. The complexity of the ground segment architecture affects processing considerations in regards to downlink bandwidths and data

latency issues. Similarly, increased onboard space vehicle memory drives design complexity, mass, and power which in turn impacts launch considerations and trades for ground segment processing. The decision regarding one segment will generally affect the other segments of the overall system [26].

Trade studies are required for the hardware and software architecture considerations. The hardware for ground segments is predominantly COTS computer systems. The possibilities for the COTS hardware such as servers, workstations, terminals, and networks must be evaluated against the requirements for processing capacity, speed, operability, reliability and maintainability, and security. Some ground functions can now be performed by COTS software products. The use of COTS must be evaluated for performance, supportability, and cost effectiveness. Software architecture involves many trades to achieve performance, accuracy, maintainability, efficiency, security, and other related goals. Other trade areas for ground software include the use of service oriented architectures (SOA), database management options, network architecture and control, and internal and external communication methodologies.

10.6.2 Concept Design Center

The Concept Design Center (CDC) is an Aerospace Corporation SE capability that uses an integrated collaborative environment to generate design and related data for complex space systems. The CDC performs major systems analysis; evaluations of launch, space, and ground systems; and provides modeled alternatives with full end-to-end scope. The models used are based on data from actual programs informed by subject matter experts' engineering judgment.

The CDC is often used to assess the cost tradeoffs associated with requirements and alternative architectures and designs. The CDC's application of integrated collaborative engineering ensures robust, quality design while dramatically reducing design time, cost, and other technical risks. Of all decisions affecting lifecycle costs, approximately 70% are made during concept design. The CDC can assist decision makers in many areas including early programmatic decision making, proposal analysis and source selection, program support, and program reduction.

The Ground Systems Team (GST) is a special CDC team that focuses on ground systems design and architecture. Customer and GST experts' pre-study collaboration is critical for a successful execution of a GST study. The GST ground architecture functions include mission management, ground command, control and telemetry, ground system management, support functions, facilities management, and operations and maintenance.

10.7 Programmatic Considerations

10.7.1 Cost and Schedule

Initial cost and schedule estimates for a ground segment are based on the total set of hardware required, which includes not only the operational hardware but also all of the hardware required for the development and test facilities, and software development through verification. Hardware costs are generally well understood, but the evolution of the technology over the total development period can produce some significant cost growth.

Historically, the major cost issues with a ground segment are the size and complexity of the software. Cost estimation tools, such as SEER-SEM can produce a reasonable cost and schedule estimate based on the estimated size of the effort and projected productivity based on a family of parameters. Generally, it has proven to be very difficult to estimate the size, especially for large (millions of lines of code) complex ground systems. To manage cost and schedule, it is necessary to develop and maintain models of the software that can be updated as size estimates are changed and used to aid replanning as necessary.

In addition to the ground system hardware and software cost estimates, additional software, hardware, and data funding will be needed for the operations and support staff to maintain the ground system, to support satellite anomaly resolution, to train the operators, to provide administrative support to the staff, and many other non-operational functions needed to support 24/7 operations.

10.7.2 Work Force

SE requires sufficiently talented and experienced program and SE managers. Seasoned leaders are critical to maintaining focus and discipline. Robust SE is critically dependent on having experienced SE with adequate knowledge of the relevant domain. Much of this domain knowledge is derived from experience by working similar programs and problems. A good systems engineer will have critical thinking and problem solving skills as a foundation, and then experiential knowledge and understanding of engineering in the context of the entire system. SE is enabled by tools developed to assist in the management of SE (not the practice of SE). Complex system development requires a full spectrum of relevant domains engaged in the earliest phases of a program that are appropriately staffed and resourced to ensure effective SE [5].

10.8 Lessons Learned

Ground segments have experienced considerable schedule, cost and technical problems in the early years of acquisition. The problems stemmed mostly from

inadequate adherence to good SE practices [27]. Some of the key recommendations based on lessons learned include4s:

- Integrate the CONOPS with requirements—functionality is important to successful development as well as verification activities
- Decompose and allocate requirements prior to key design decisions
- Anticipate that requirements will change; planning for change is essential
- Establish well-defined system and software architectures
- Develop thorough and complete test plans, as well as adequate test facilities
- Factor HSI into the requirements and design
- Include margin in schedules to account for program complexities (e.g., growth), resource limitations, and external dependencies
- Ensure cost performance (earned value) is consistent with the program's technical maturity. Cost performance should provide a true indicator of program direction
- Ensure the schedule supports the processes
- Enforce agreed to content and criteria
- Institutionalize cost and schedule processes

Similar findings are documented in the Aerospace report Lessons Learned from Independent Program Assessments [28]. Some of these findings are:

- Poor government cost baseline (e.g., awarding the acquisition contract based on less than government cost estimate)
- Poor schedule baseline (e.g., awarding the acquisition contract based on a schedule shorter than government schedule estimate),
- Not using technology on/off ramps effectively
- Changes in major requirements after acquisition contract award
- Poor government program office technical baseline
- Poor contractor processes and poor implementation of those processes
- Poor government oversight of contractor processes and testing
- Program disruption due to problems in government decisions
- Too much time required for request for proposal preparation and source selection
- Too many budget cut drills
- Difficulties in meeting obligation and expenditure standards, resulting in budget cuts
- Other SE shortfalls include T&E planning, requirements decomposition and traceability, trades, interface planning
- Government program office not applying the lessons learned and best practices derived from past program experience

- Too few qualified people in the government program office and contractors

10.9 Summary

Application of formal SE processes throughout the acquisition lifecycle of a ground segment is critical to fielding the required system on time and on budget. Correct and accurate requirements definition appropriately allocated will ensure a more robust and stable technical baseline whether the segment is being acquired as part of a system or is independently acquired. SE's greatest successes will be achieved by ensuring all ground segment interfaces, particularly external, complexities are well identified and minimized. SE processes and products are applicable to all aspects of the segment development; however the application to the segment/system functions will affect implementation of process and required products as well the oversight role of the acquisition authority team in the number and sequencing of technical and independent reviews and assessments. Effective application of SE process, tools, and methodologies are only as good as the system engineers (e.g. experience and aptitude) that apply and execute from an inter-disciplinary perspective across the segment elements.

10.10 References

1. Office of the Deputy Under Secretary of Defense for Acquisition and Technology. *Systems Engineering Plan Preparation Guide*, Version 2.01. Department of Defense. Washington, DC.April 2008.

2. Blanchard, Benjamin and Wolter Fabrycky, *Systems Engineering and Analysis* (4th Edition), Englewood Cliffs, N.J.: Prentice Hall. 2005.

3. Modified from Systems Design and Operational Effectiveness 625 Class Note—"Systems Design and Operational Effectiveness," Stevens Institute of Technology, 2007.

4. INCOSE Systems Architecture Tools Survey, available at http://www.paper-review.com/tools/sas/index.php. Last accessed on April 2, 2007.

5. Pre-Milestone A and Early-Phase Systems Engineering: A Retrospective Review and Benefits for Future Air Force Acquisition, http://www.nap.edu/catalog/12065.html.

6. Shaw, Brian E. *Systems Engineering Requirements and Products*. TR-RS-2013-00001, The Aerospace Corporation, El Segundo, CA. February 28, 2013.

7. International Council on Systems Engineering (INCOSE), 2004, *INCOSE Systems Engineering Handbook (Version 2A)*, Seattle, Wash.: INCOSE.

8. Guarro, S. B., G.A. Johnson-Roth, W.F. Tosney. *Mission Assurance Guide.* TOR-2007(8546)-6018 Rev B. The Aerospace Corporation, El Segundo, CA. June 2012.

9. Tosney, W.F., P.G. Cheng, and J. J. Juranek. *Guidelines for Space Systems Critical Gated Events*, TOR-2009(8583)-8545. The Aerospace Corporation, El Segundo, CA. May 9, 2008.

10. Perestegy, L.B. and C.E. O'Conner. *Technical Reviews and Audits for Systems, Equipment, and Computer Software*. TR-RS-2009-00021, The Aerospace Corporation, El Segundo, CA. September 15, 2009.

11. Covington. R. K. *Design Review Improvement Recommendations*. TOR-2015-02545, The Aerospace Corporation, El Segundo, CA. June 18, 2015.

12. Lutton, David A., Colleen M. Ellis, James A. Sheer, Suellen Esslinger, and Brian E. Shaw. *Proposed Test Requirements for Ground Systems*. TOR-2013-00175. The Aerospace Corporation, El Segundo, CA. 2013.

13. White, J. and L. Tilney. *The Test Like you Fly Process Guide for Space, Launch, and Ground Systems*. TOR-2014-02537, The Aerospace Corporation, El Segundo, CA. July 28, 2014.

14. http://acqnotes.com/acqnote/careerfields/systems-engineering-planse.

15. http://acqnotes.com/acqnote/acquisitions/initial-capabilities-document-icd.

16. http://acqnotes.com/acqnote/acquisitions/acquisition-strategy.

17. http://acqnotes.com/acqnote/careerfields/systems-engineering-management-plan-semp.

18. Larman Craig. *Agile and Iterative Development: A Manager's Guide*, Addision-Wesley. 2003.

19. Software Engineering Institute. CMMI for Development, Version 1.3 (CMMI-DEV) CMU/SEI-2010-TR-0033, November 2010.

20. Eslinger, Suellen. *Software Acquisition, Software Engineering, and the Aerospace Role*. The Aerospace Corporation, El Segundo, CA. 2006.

21. Donahue, C. and B. McKinzey. *Configuration Management*, TOR-2006(8583)-1, The Aerospace Corporation, El Segundo, CA. August 2005.

22. *Space Vehicle Systems Engineering Handbook*, Major Interfaces chapter. TOR-2006(8506)-4494, The Aerospace Corporation, El Segundo, CA.

23. Department of Defense Instruction. *Cybersecurity*. 8500.01. March 14, 2014.

24. Ross, Ron, Stu Katzke, Arnold Johnson, Marianne Swanson, Gary Stoneburner, George Rogers, and Annabelle Lee. *Security and Privacy Controls for Federal Information Systems and Organizations*. NIST 800-53. National Institute of Standards and Technology. Gaithersburg, VA. February 2005.

25. Chien, Jya-Syin W. and James Donndelinger. *Introducing Cybersecurity into the FMECA Process for National Security Systems*, ATR-2016-00605, The Aerospace Corporation, El Segundo, CA. December 4, 2015.

26. Wheaton, Merilee J. An Overview of Ground Systems Operations. *Crosslink*, The Aerospace Corporation, El Segundo, CA. Spring 2006.

27. Gomez, A. and R. Sudakow. *Application of SBIRS Lessons Learned to HEO and GEO Development Activities*. TOR-2004(1472)-3533e, The Aerospace Corporation, El Segundo, CA. June 24, 2004.

28. Bonesteele, R. *Lessons Learned from Independent Program Assessments*. TOR-2007(8506)-7167, The Aerospace Corporation, El Segundo, CA. September 20, 2007.

10.11 Bibliography

SMC Systems Engineering Primer & Handbook, 11 March 2013

IEEE 15288.1, Standard for Application of Systems Engineering on Defense Programs, 2014

ISO/IEC 15288, Systems Engineering-System Life Cycle Processes

ANSI/EIA 632, Processes for Engineering a System

IEEE 1220, Application and Management of the Systems Engineering Process

EIA 731, Systems Engineering Capability Model

CMMI, Capability Maturity Model Integration

G.S. Arnold et al. Mission Assurance Baseline (MAB) Version 2.6, ATR-2014-00483, The Aerospace Corporation, El Segundo, CA. 2013.

Defense Acquisition Guidebook, Chapter 4

AFI 63-1201, Life Cycle Systems Engineering

IEEE/EIA 12207, Software Life Cycle Processes

Air Force Weapon System Software Management Guidebook

Naval Air Systems Command, NAVAIR, Systems Engineering Guide, May 2003

10.12 Acronyms

AoA	analysis of alternatives
CBA	capabilities based assessment
CCB	configuration control boards
CDC	Concept Design Center
CDD	capability development document
CDRLs	contract data requirements lists
CI	configuration item
CONOPS	concepts of operation
COTS	commercial off-the-shelf
CPD	capability production document
CWBS	contract work breakdown structure
DAL	data acquisition list
DM	data management
DOD	Department of Defense
DOORS	dynamic object oriented requirements system
DT&E	developmental test and evaluation
EOT	early orbit test
EVMS	earned value management system
FFRDC	federally funded research and development center
GST	ground systems team
HSI	human systems integration
ICDs	interface control documents
IMP	integrated management plan
IMS	integrated management schedule
MAB	mission assurance baseline
MDD	material development decision

OSD	Office of the Secretary of Defense
OT&E	operational test and evaluation
RFP	request for proposal
SDP	software development plan
SE	systems engineering
SEER-SEM	systems evaluation and estimation resources-software engineering model
SEIT	system engineering integration team
SEMP	systems engineering management plan
SEP	systems engineering plan
SETA	system engineering technical assistance
SOA	service oriented architectures
STP	system test plans
T&E	test and evaluation
TEMP	test and evaluation master plan
TLYF	test-like-you-fly
TPM	technical performance measures
WBS	work breakdown structure

Chapter 11
Ground Segment Software

Leslie J. Holloway (retired)
Computers and Software Division
Karen Owens (retired)
Computers and Software Division
Martha I. Johnson
Software Acquisition and Modeling Department
Software Engineering Subdivision

11.1 Introduction

Ground systems acquired as part of national security space (NSS) systems typically contain millions, or even tens of millions, of lines of software. Many of these acquisitions have had extensive software-related problems that negatively affect system cost, schedule, performance, and quality. As a result, it is essential that decision makers should carefully consider software in every aspect of a ground segment acquisition.

This chapter provides an overview of the acquisition of ground segment software. It includes discussion of the software acquisition lifecycle, the software development lifecycle, and requirements for software development. It encompasses the acquisition lifecycle from early technology development through requirements, architecture, design, implementation, integration, test, deployment, operations, and sustainment. It discusses software processes, software products, and software reviews. It focuses on the role of the government software acquisition team in specifying software requirements to the contractors, in acquiring software as part of a ground segment, and in monitoring the performance of the developing contractors as they design, implement, integrate, test, and deploy the software. In addition, best practices for each of the software acquisition activities are provided.

Note: Throughout this chapter, the term "software acquisition team" is used to represent the team composed of the program's Chief Software Engineer (or equivalent) and support personnel, generally consisting of government, Federally Funded Research and Development Center (FFRDC), Systems Engineering and Technical Assistance (SETA), and Systems Engineering and Integration (SE&I) contractors.

11.2 Definitions

Acquisition The process of obtaining products or services through documented supplier agreements. (e.g., contracts, licenses), [1]

CMMI® The Capability Maturity Model Integration models are collections of best practices that help organizations to improve their processes. Best practices in the model focus on activities for developing quality products and services to meet the needs of customers and end users. In addition to the models, there is a basic framework that provides the models, appraisal methods, and training materials. [1, 2]

Contractor A person or organization that enters into a contract with the program for the supply of a product. The term "contractor" is intended to provide a neutral and broad definition of acquisition that includes those delivering products or performing services as well as those contracted (such as a prime contractor) to develop and deliver products.

Lifecycle Evolution of a system, product, service, project, or other human-made entity from conception through retirement. [3]

Peer review The review of work products performed by peers during the development of the work products to identify defects for removal. [2]

Product A work product that is intended for delivery to a customer or end user. [2]

Software acquisition The set of processes (e.g., the set of policies, procedures, methodologies, tools) used by the Acquisition Program Office (APO) to acquire software.

Software acquisition team The team composed of the program's Chief Software Engineer and support personnel, generally consisting of Air Force or Government, Government Civilian, and FFRDC, SETA and SE&I members.

Software engineering The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. [4]

Software process An organized and integrated set of activities whose purpose is to translate user needs into software products. It transforms inputs into outputs, to achieve a given purpose. It includes the means by which the activities are accomplished; that is, the methods, techniques, practices tools, standards, resources, etc. It includes the interrelationships among the software activities themselves and between the software activities and other systems development and management activities (e.g., systems engineering). [2]

Work product The useful output or result of performing a process. This result could include files, documents products, parts of a product, services, process descriptions, specifications, models, or code. A key distinction between a work

product and a product component is that a work product is not necessarily part of the end product. [2]

11.3 Software Lifecycle Acquisition

Often software does not have a large role in the early phases of the acquisition lifecycle; however it is important that software be represented in many of the early lifecycle activities, such as developing the technical requirements for the system, developing the cost estimates to support the budgeting process, generating the request for proposal (RFP), and performing source selection.

A large ground segment will go through several contract actions through the acquisition lifecycle. There are often separate contracts for each ground segment acquisition lifecycle phase. These phases generally include a technology maturation and risk reduction (TMRR) phase, an engineering and manufacturing development (EMD) phase, a production and deployment (P&D) phase, and an operations and support (O&S) phase. It is important that software personnel contribute to the RFP development, source selection, and contract preparation for each of these phases.

The ground segment contractors are responsible for planning the software development, selecting a software development lifecycle model, defining the processes to be used, performing to the defined plans, processes, standards, ultimately producing quality products. The government software acquisition team has an oversight role in each of these activities, ensuring that the contractors adhere to the defined processes, perform to the planned schedule, and produce products of acceptable quality.

In addition to the product-oriented software development activities (requirements, architecture, design, implementation, integration, and test) that result in the production of the product there are many activities that the contractors perform in support of these mainline activities. Software is an important topic in the system level technical reviews. The most common name(s) for these reviews are: system requirements review (SRR), system design review (SDR) [sometime known as system functional review (SFR)], software requirements and architecture review (SAR), preliminary design review (PDR), critical design review (CDR), and test readiness review (TRR). Software should also be an important topic in program management reviews. Software may be deployed several times during a ground segment acquisition. Some software deployments will be internal to the contractor — to a modeling and simulation risk reduction activity, or an early systems integration and test activity, for example. Some software deployments will be from the contractor to a government site — for early operational testing or for interface testing, for example.

Once software has been deployed, whether to an internal or external entity, it must be maintained in that environment. Maintenance includes upgrades to the software to fix defects, improve performance, add features, or refresh technology.

11.3.1 Software in the Early Acquisition Lifecycle

After the decision has been made that a system is to be acquired, a program often starts with a TMRR phase. The purpose of this phase is to mature the high-risk technologies, to determine if the system can be built within cost and schedule constraints, and to evaluate the capability of competing contractors. Often, software is not considered one of the high-risk technology areas, and frequently there are no software personnel in the acquisition program office during this phase. This is a mistake. Software is almost always a major component of a ground segment acquisition, and the sheer volume of software to be developed or acquired makes managing the software acquisition a risk.

Software personnel need to be involved in the early acquisition phases to ensure that the requirements and architecture developed for the system accurately reflect the amount of software to be developed and maintained, to define what the software is capable of doing, to identify the software related risks, and to plan how the software is to be developed, integrated, and tested.

In support of the budgeting process, the acquisition program office must generate an estimate of the cost of acquiring and maintaining the system (full acquisition lifecycle cost). Because the cost of the software development will be a large component of the ground segment cost, it is important that the government software acquisition team accurately estimate the amount of software to be developed or acquired. This requires the participation of software subject matter experts who are familiar with the functionality of the system, with the available non-developmental item software [commercial off-the-shelf (COTS), government off-the-shelf (GOTS), “Reuse” software, open source software (OSS)], and with the complexity of the software to be developed. The subject matter experts must develop a high-level architecture of the software, identify critical interfaces (internal and external), and be able to estimate the amount of software to be developed and maintained. The size estimates are used to provide cost and schedule estimates for the program, using commercial or government cost estimating tools. It is important that these estimates be independent of any estimates provided by potential developing contractors.

The technical requirements for the system are derived from the user’s mission needs. Because many of the system requirements will ultimately be implemented in software, it is important that software personnel be involved in generating and reviewing the technical requirements document. For example, many ground systems have an availability requirement, that is, the proportion of

time that the system must be operational. Because software can certainly affect whether or not the system is operational, software must be reflected in the availability requirement. In the past, many systems assumed that the availability of software was 1.0, because there were no effective methods for estimating software availability, reliability, or maintainability. This is no longer true.

There are additional non-functional requirements in the technical requirements, such as safety, human systems integration, flexibility, extensibility, maintainability, software assurance, and cyber security that are also at least partially implemented in software. It is important that software personnel be involved in generating and reviewing these requirements to ensure that they are stated in such a way that they can be effectively implemented in software.

11.3.2 Software Topics in Contracting

Almost all acquisitions of NSS system ground segments include the acquisition of software. It is therefore important that software acquisition personnel be involved in preparing the RFP, participating in the source selections, and contributing to the contract clauses. Software contract elements include Section B provisions for data rights, software maintenance support, and software operations support. Section H provisions for software include data rights purchase options, earned value for software, software architecture evaluations, software process appraisals, software build reviews, software readiness assessments, and retention of key software personnel. Section J provisions for software include software-related attachments to the proposal, recommended software contract data requirements list (CDRL) items, the data accession list (DAL), award and incentive fees, the compliance and reference documents list, the integrated management plan (IMP), key personnel positions, government-furnished property, data rights, the statement of work (SOW), the system requirements document, and the work breakdown structure (WBS).

The proposal instructions in Section L and the evaluation criteria in Section M should reflect the importance of software in the ground segment acquisition, enable evaluation of the software risks to the technical approach, and allow the evaluation of processes used or proposed to be used.

This section addresses some specific software considerations in contracting. Additional guidance can be found in Aerospace Report No. TOR-2011(8506)-117, *Integrating Software Topics into the Request for Proposal* [5], hereafter referred to as the RFP TOR.

11.3.2.1 Uniform Contract Format Section B

Section B of the uniform contract format contains a description of each contract deliverable covered by a contract line item number. There is a contract line item

for every item supplied by the contractor to the government, including every delivery of software. Of particular interest for software are the data rights associated with these software deliveries and the sustainment of delivered software.

Data rights are important in software because the government software acquisition team needs the rights to the data to operate, support, and maintain the system. Some data rights come at no additional cost with the contracted system while some must be purchased. Some come without restrictions, some come with restrictions and license issues that must be dealt with. More details for license issues are discussed in more detail in Aerospace Report No. TOR-2014-02681, *General Guidance on Open Source Software* [6], hereafter referred to as the OSS TOR. When the government does not want to purchase the rights to software and technical data initially, but does want the option to purchase them later, it is more cost effective to negotiate the price of those rights during the initial contract negotiations. The vehicle to do this is to identify data rights for contract data requirements list items as contract line items in Section B of the contract. This provides the government options that may be exercised at a later time. More details and specific contract language for rights in technical data and software are discussed in more detail in the RFP TOR [5].

The government software acquisition team must specify when it wants to purchase maintenance and operations support for delivered software. This may be done multiple times during the ground segment lifecycle if the software has multiple deliveries. This is done by defining a contract line item to deliver support services or maintenance services for each software delivery. See the RFP TOR [5] for more details on this topic.

11.3.2.2 Uniform Contract Format Section H

Section H of the uniform contract format contains special contract requirements. Of particular interest for software are data rights, earned value management, the FFRDC enabling clause, software architecture evaluations, software process appraisals, and retention of key software personnel.

To allow for the deferred purchase of data rights in Section H, the data rights must be specified as a contract line item in Section B as discussed above. Specific Section H language for data rights options is provided in the RFP TOR [5].

Defense acquisition regulation supplements [7] require the use of an earned-value management system by contractors. Often, earned-value data are only required to the third level of the WBS. This provides metrics at the system, segment, and subsystem level. This is not sufficient for software, which typically resides at the fifth or sixth level of the WBS. The RFP and contract, therefore, must specifically require the contractors to report software metrics at

the required level. The minimum acceptable level is at the software item level, but additional levels may be required for specific programs. Therefore, it is necessary to specify the WBS level for collecting and reporting earned-value data for software in the SOW. See the discussion in the RFP TOR [5].

In order for FFRDC personnel to obtain data from contractors, there must be an enabling clause in the contract. This enabling clause describes the relationship between the government and the FFRDC, describes the FFRDC role in general systems engineering and integration, and requires the contractor to provide both cost and technical data to FFRDC's personnel. This is especially important for software personnel to obtain access to the many software products produced over the software development lifecycle and to the tools that generate these products.

Of particular interest in a ground segment is the software architecture. The software architecture implements the non-functional requirements of the system (reliability, availability, maintainability, extensibility, human systems integration, safety, cyber security, etc.) and is critical to the operational performance of the system. In order to ensure that the contractor's software architecture meets program requirements, the contractor should be required to conduct periodic software architecture evaluations. In addition, the government software acquisition team should be granted the right to conduct periodic independent software architecture evaluations. The recommended methods for performing software architecture evaluations are either using the Software Engineering Institute's Architecture Tradeoff Analysis Method [8] or The Aerospace Corporation's software architecture evaluation method [9]. Section H language to enable these activities is provided in the RFP TOR [5].

Because "the quality of software is largely determined by the quality of the process used to develop and maintain it" [10, page 8], it is important that the government software acquisition team be able to conduct periodic process appraisals[3] on the contractor. The special contract requirement should: require the contractor to perform periodic process self-appraisals, allow the government software acquisition team to perform periodic independent appraisals, require the contractor to improve processes continuously, require the contractor to document identified weaknesses and risks in a process improvement plan, and

[3] SM CMM Integration, SCAMPI, SCAMPI Lead Appraiser are service marks of Carnegie Mellon University. The industry standard for performing process appraisals is the use of the Standard CMMI® Appraisal Method for Process Improvement (SCAMPI℠), using CMMI® and a CMMI® Institute Certified Lead Appraiser. For more information on CMU/SEI Trademark use, please visit http://www.sei.cmu.edu/about/legal-trademarks.html [11].

require the contractor to flow down this contract requirement to all subcontractors. Recommended Section H language is provided in the RFP TOR [5].

Many RFPs require that the offerors submit an attachment describing the qualifications of some of their key personnel. Section H is where the government may specify the personnel positions that are considered key. For software, some key personnel may be the software integrated-product team lead, the software chief engineer, the software chief architect, and the software security specialist. An example of language for key personnel retention is provided in the RFP TOR [5].

11.3.2.3 Uniform Contract Format Section J

Section J of the RFP specifies the required attachments to the proposal. Many of these documents contain significant software content. These often include the requirements for: the CDRL, DAL, award or incentive fee plan, compliance and reference documents list, integrated master plan, key personnel positions, list of government furnish property, rights in technical data and software attachment, SOW, system requirements document, and WBS.

The CDRL identifies the product deliverables that the contractor will provide to the government during the execution of the program. The deliverable products and data on the CDRL are referred to as CDRL items. They consist of documents, source code, plans, schedules, drawings, and other types of technical and management information. Two Aerospace TORs identify recommended software-related CDRL items [12] and [13]. The *Software Development Standard for Mission Critical Systems* [14] (hereafter referred to as the SDSMCS) lists the recommended data item descriptions for each of the recommended CDRL items, as well as guidance on delivery and tailoring. Since the publication of [12] and [13], updates have been made to the recommended software deliverables to accommodate modern software development. These are identified in the SDSMCS and recommended content and format are provided.

In addition to CDRL items, a number of products are prepared by the contractor that are used to develop, test, and manage the program that the contractor is not required to deliver to, or obtain approval from, the government. A CDRL item called the DAL consists of an index of these non-deliverable types of items. Some of the items that may be considered for DAL are: meeting minutes, metrics reports, software unit test cases and results, quality assurance process and product audit results, and schedules maintained below the deliverable schedule level. The government can request access to items on the DAL through the electronic data interchange network. The government software acquisition team should ensure that the software products of interest are included in the DAL.

In many acquisitions, the government provides an award or incentive fee plan with the RFP, with the offerors providing a more complete award or incentive fee plan with their proposal. Because software is such a large component of a ground segment it is essential that software personnel with appropriate expertise participate in the development of the award or incentive fee structure. A structured risk analysis should be performed to determine an award or incentive fee structure that balances cost, schedule, and quality of the software. Software personnel should review the contractor's proposed award or incentive fee plans to ensure that the importance of software is adequately reflected in the plan.

The RFP contains a compliance and reference documents list that itemizes all compliance and reference documents specified in the SOW and the technical requirements document (TRD). A government software acquisition team should ensure that as many recommended software-related compliance and reference documents are included as are appropriate for the acquisition. The most important software compliance documents are listed in the SDSMCS [15] and the *Software Measurement Standard for Space Systems* [16] (hereafter called the Software Measurement Standard). The SDSMCS specifies the minimum requirements for effective software development. The Software Measurement Standard specifies the required metrics for managing the software aspects of the program. By implementing these metrics, the government has visibility into software development processes and products throughout the lifecycle. Additional software compliance and reference documents that may be appropriate to a ground segment acquisition are identified in the RFP TOR [5]. Because acquisition rules and policy change frequently, the reader is encouraged to consult current policy and software acquisition subject matter experts when developing the compliance and reference documents list. Tailoring of the standards and CDRLs and adding program unique standards may be necessary for a particular acquisition.

An important attachment to the proposal is the government's IMP. The IMP is an event-based, high-level plan for managing program progress. The government software acquisition team defines the major program events in the IMP; in the proposal, the contractor elaborates the IMP events with significant accomplishments (lower-level tasks that achieve the event) and accomplishment criteria (the criteria for determining whether the significant accomplishment is achieved). It is important that software personnel ensure that software-specific events are defined in both the government and contractor IMPs. Software-specific events may include a SAR, defined in [17] and [18]. Other software events and accomplishments may include software architecture reviews, process appraisals, software readiness assessments, and software build-level reviews.

Another frequent attachment to the proposal is a description of the key personnel on the program, their qualifications for the position, and how the contractor intends to motivate and retain them. The government software acquisition team

should review the qualifications for the key software positions to ensure that the contractor are qualified personnel. Key software positions may include the software integrated product team (IPT) lead, chief software architect, chief software engineer, and software security specialist.

Government-furnished property (GFP) (including government-furnished information) may be required by the program or requested by the contractor. There are many reasons why GFP may be required or requested: the system needs to interface with the GFP, the system needs to embed the GFP, or the system needs to be hosted on the same hardware as the GFP. The government software acquisition team needs to identify what software-related GFP needs to be included in the RFP. See the RFP TOR for additional information about software-related GFP [5].

The data rights to be supplied by the offeror are identified in an attachment for rights in technical data and software. The offered data rights must meet legal constraints and the government's requirements specified in the RFP; there may be additional rights to enhance the offeror's competitive position. Data rights are an important topic for software and there should be subject matter experts involved in the preparation of the RFP and reviewing the proposal attachment to ensure that the government is getting the best value for the data rights that it needs to operate, sustain, and maintain the system for the entire operational life. The RFP TOR discusses data rights in some detail [5].

The SOW specifies each task the contractor performs. There must be a SOW task for every CDRL item and each SOW task must result in a work product that may be a CDRL item, a DAL item, or some other work product. Some documents, such as the Software Development Plan (SDP), are sufficiently important that the government wants to ensure that the contractor generates a plan that details their processes and procedures for performing software development, and then actually performs according to the plans in that document. The SOW provides the government the opportunity to make these documents contractually binding by using the phrase "the contractor shall perform these tasks in accordance with the plan." This is specifically recommended for the SDP and may be useful for other software products as well. It is very important for the government software acquisition team to review both the government and contractor SOWs to ensure that all the necessary software tasks for the program are specified, along with the appropriate work products. See the RFP TOR for additional software specific SOW tasks [5].

Many ground segment programs for national security space systems use the WBS defined in MIL-HDBK-881, *Work Breakdown Structures for Defense Materiel Items* [19], which is based on integrated product development. In this handbook software is contained within each ground element, often at the fifth or sixth level of the WBS. This provides minimal visibility into the contractor's

software development activities, risks, and status. Based on the increasing importance of software in NSS systems, Aerospace has developed a technical report [20] that recommends that software have its own common element at level 2 in the WBS, parallel with the systems engineering common element. The recommended WBS also shows how software should be included through the fourth level of the WBS.

11.3.2.4 Uniform Contract Format Sections L and M

Section L provides the instructions to offerors for how to prepare their proposals and Section M provides the evaluation criteria for their submitted proposals. It is important that the instructions and evaluation criteria accurately reflect the importance of software in the ground segment to be acquired. The government software acquisition team must be actively involved in the preparation of the RFP and in the evaluation of the contractors' proposals, along with subject matter experts for specific software-related topics. The RFP TOR [5] provides information to assist software personnel in ensuring that software is adequately covered in the proposal instructions and evaluation criteria.

11.3.3 Software Development Lifecycle

The primary document governing the software development is the SDP. This document is prepared by the contractors, and is often delivered for the first time with the proposal and then updated as needed throughout the program's lifecycle. The SDP defines the software development lifecycle model, software development processes and work products, and software development environment that the contractor intends to use for the program and the associated project product standards. The requirements for the SDP are specified in the SDSMCS [14], which is recommended to be a required document for every NSS system program.

The software development lifecycle model should be selected for the program based on the requirements, scope, and complexity of the software. Typical software development lifecycle models include waterfall, incremental, iterative, evolutionary, and spiral. These lifecycle development models are described in the Aerospace Report No. TOR-2013-00692, *Iterative Software Development in Space Systems Acquisition* [21]. Additionally, the contractor may propose a development method, such as agile or unified process. The government software acquisition team should carefully review the contractor's proposed lifecycle model and methods to ensure that these are consistent with the requirements and constraints of the program, and that the contractor has sufficient experience with the proposed model and methods for a successful execution.

The software development processes should be selected for the program by tailoring the contractor's organizational set of standard procedures. If the contractor has a team developing the software, the processes must be integrated across the team. Early in the life of the program the contractor should be subject to a process appraisal, either a self-appraisal by the contractor with government participation, or (preferred) an independent appraisal performed by the government. The appraisal determines the maturity of the documented processes across the team and whether (1) during the proposal, the proposed processes have been used effectively on other programs and will be effective on the subject program by all team members, and (2) during contract performance, the program personnel throughout the team are actually following the documented processes.

As discussed above, at appropriate milestones, the software architecture should be evaluated. This can be either a self-evaluation by the contractor with government participation, or (preferred) an independent evaluation performed by the government.

An important topic in determining the scope of the software development is the estimate of the amount of developed software, reuse software, COTS software, and open OSS. The SDP should itemize each of these sources of software, identify the product selected, and provide the rationale for the selection (see the SDSMCS, Appendix B [14] for more detail). The government software acquisition team should carefully review the proposed software products, with special attention given to the security implications of reuse, COTS, and OSS. The contractor should provide a plan (e.g., technology refresh plan, software sustainment plan) for how each of the products will be transitioned to operations, sustainment, and maintenance and maintained for the operational life of the system.

Another important topic in the SDP is the definition of the software engineering environment [defined to include the software development environment, software integration and test environment, and software configuration management (SCM) environment]. The government software acquisition team should carefully review the contractor's plans for the design, implementation, sustainment, and maintenance of the software engineering environment. It is important for the government software acquisition team to determine if the software engineering environment is adequate, and is implemented in a timely manner across all software development organizations. Additionally, there should be plans to maintain (and refresh when needed) the software engineering environment throughout the software development lifecycle.

The SDSMCS has many requirements for the verification of software, at the unit, software, hardware-software integration, and system level. The government software acquisition team should review the contractor's software test plans for completeness and accuracy. Unit test plans should include software tests for

nominal, off-nominal, and stressing cases. Both functional and non-functional (quality attribute) requirements should be verified. The test plans should make clear when and where software is verified and when and where both the functional and non-functional requirements are verified. The government software acquisition team should ensure that adequate test facilities have been provided across the software development team. The government software acquisition team should ensure that the appropriate verification method (inspection, test, demonstration, or analysis) is identified for each requirement, and that adequate validation modeling and simulation is performed to verify performance requirements.

An important component of government oversight of the contractor's software development activities is the use of independent analyses. There are subject matter experts within the FFRDCs for every conceivable software topic. The government should verify the quality of the contractor's software development activities through the judicious use of independent analyses by subject matter experts.

11.3.4 Software Support Activities

The SDSMCS also defines requirements for the software activities that are not mainstream to product development, but support product development. These include:

- Software project management (project planning, project monitoring and control, management reviews, measurement and analysis, risk management, corrective action, configuration management (CM) and subcontractor management);
- Software process management (process definition and process improvement);
- Specialty engineering [reliability/maintainability/availability (RMA)], safety, security (software assurance and cyber security), human systems integration (HSI), supportability, fault management, integrated hardware (HW)/software (SW) diagnostics, COTS software supportability]; and
- Software quality enhancement (technical reviews, product evaluations, peer reviews, and quality assurance).

Software project management includes all the activities that the software development contractor performs to plan, monitor, and control the software portions of the program. The Capability Maturity Model for Development

(CMMI®-DEV)[4] [2] describes the characteristics of good project management and product development.

The contractor should plan all software development activities; this not a one-time activity, but an ongoing process throughout the software development lifecycle. Planning includes quantitative planning (software cost, schedule, staffing, and other resources) and qualitative planning (software development activities, processes, methods techniques, procedures, product standards, and work instructions). The planning includes what is to be done, who performs it, how is it performed, where is it performed, what is the scope of the task, when it is to be performed, why it is performed, and identifying any dependencies between tasks. The planning products include the SDP, software size estimates, software processes, software architecture, and software WBS. The government software acquisition team should ensure that all of the necessary planning products are produced and they represent an executable program. Independent analyses may be performed to validate contractor plans.

The contractor must also plan all software development processes. Again, this is not a one-time activity, but must be revised as processes are modified and improved. Software processes are documented in the SDP and often reference other process documents. The government software acquisition team should ensure that all of the necessary process products are produced and are executable and verifiable. In particular, integration of processes across the software development team is important. The ETVX (entry, task, verification, exit) approach is a recommended method for documenting processes for software development activities; see the SDSMCS [15, Appendix H]. Independent analyses may be performed to validate contractor process plans.

Once there is a plan in place software development progress can be measured against the plan. The Software Measurement Standard [16] provides recommended metrics for software development across the lifecycle. The contractor should define, collect, analyze, and report software metrics that measure whether: the project is on schedule, costs are under control, requirements are stable, defects are being corrected, training is being delivered on time, and staffing is sufficient. Additional metrics may be defined to address specific risks, concerns, and issues. The government software acquisition team may perform independent analyses to validate the contractor's reported progress.

[4] CMMI®, Capability Maturity Model, Capability Maturity Modeling, CMM, and Carnegie Mellon are registered in the US Patent and Trademark Office by Carnegie Mellon University. For more information on CMU/SEI Trademark use, please visit http://www.sei.cmu.edu/about/legal-trademarks.html [2].

A software risk management process must be defined and practiced by all software development organizations at all levels of the IPT hierarchy and by software personnel at all levels. The software risk management process must be integrated with the program-level risk management process. Software risks with significant consequences must be elevated up the IPT hierarchy, especially when the risks cross contract boundaries. Everyone in the program organization needs to own the management of software risks. Identifying risks and providing suggestions for risk mitigation must be rewarded throughout the organization. The government must maintain a risk management process independent of the contractor's.

11.3.5 Software Reviews

Opportunities for evaluating the status of software development are provided at a variety of software reviews. The software architecture evaluation and process appraisal were discussed previously. In addition, there are technical reviews, management reviews, and product reviews.

Technical reviews include the system level reviews (SRR, SDR/SFR, SAR, PDR, CDR, and TRR) and software build-level reviews. Mission success criteria of the system-level reviews are defined in Aerospace documentation [17, 18]. The software component of the system-level reviews is discussed in some detail, including how to review the status of software in an iterative software development. The SDSMCS defines the scope of software build-level reviews [14]. The government software acquisition team must ensure that the contractor adequately plans and executes the necessary software reviews over the lifecycle of the software and system development. The government is responsible for ensuring that the contractor is not allowed to pass a review that does not adequately cover software.

Software must also be represented in regular management reviews. This might be accomplished by having software as a topic in monthly management reviews, conducting regular status meetings within the software organization, or both. The government software acquisition team must ensure that software status is reported, software issues identified, and software issues resolved.

Software product reviews include peer reviews and readiness assessments. The purpose of a peer review is to identify defects for removal. Peers review a work product during development. Peers are stakeholders in the process, who are affected by the work product or the process or in some way accountable for the outcome of the product or the process (excluding managers). Frequently, the government has government or FFRDC technical personnel participate in the peer reviews, both at the contractor and subcontractor level. Peer reviews may be performed on documents (e.g., plans, requirements specification, test plan, design description), on work products, on models, on process descriptions, or on

code. The most important software work products to review are the requirements and design work products. This is because defects found early in the software development lifecycle are significantly cheaper to fix than defects found later in the lifecycle. Each product is evaluated against a set of evaluation criteria defined for that specific work product and adapted to the program. Peer reviews are the most effective way to reduce rework and to ensure the development and delivery of high-quality software products. The SDSMCS [14] defines a minimum set of evaluation criteria for software products.

A software readiness assessment (also called a "gating engineering review") is an evaluation of software technical and management maturity at predefined points in the software product lifecycle. The purpose of a software readiness assessment is to determine the readiness of the software for the next phase of development, or for an upcoming designated event, and the associated risk of proceeding. The ultimate goal is to provide objective, independent feedback to the program's software stakeholders.

Aerospace Report No. TOR-2011(8591)-20, *Space Segment Software Readiness Assessment* [22], defines a process for conducting software readiness assessments and a set of assessment criteria for evaluating software readiness. In performing a software readiness assessment, the software readiness assessment team evaluates risk by assessing software planning, technical performance, execution progress, and software quality, and their effect on software cost and schedule. To accomplish this, the software readiness assessment team applies the assessment criteria provided in [22] to three perspectives of the software development program as it evolves through the lifecycle: products, processes, and resources.

If written into the contract, the government software acquisition team may perform software readiness assessments, either at pre-determined milestones or on an as-needed basis. In an iterative software development the contractor and the government should negotiate the number and schedule of the software readiness assessments soon after authority to proceed (ATP), possibly as part of an integrated baseline review (IBR).

11.3.6 Software Transition to Operations

The software development contractor should have a plan for how software will be deployed into operations, both internal deployments and external deployments. These deployments may occur multiple times during the software development lifecycle. If the contractor is using an iterative software development lifecycle model, the Master Software Build Plan (MSBP) should identify, for each iteration, what software, if any, is to be deployed.

Preparing for software use in each iteration includes: preparing the executable software, preparing version descriptions for each deployment site, preparing user manuals (software user manuals and computer operation manuals), and installation instructions. See the SDSMCS [14] for more detail about the requirements for software transition to operations.

When the system under development is replacing an existing system (in whole or in part), careful attention should be paid to how the new software in the system will be verified against the existing system to ensure backwards-compatibility requirements are met. In addition to the transition planning required for new systems, planning should include:

- An approach to run parallel operations to determine compatibility. Planning for this is essential to avoid potential conflicts.
- An approach to falling back to the prior system software if the new software fails severely.
- Where practical, plan for the new system software to be tested with live satellites as part of pre-deployment testing and transition to operations.

Legacy concerns with the existing system, such as contracting, sustainment, liability, licensing, data rights, etc. must be considered during planning and contracting. The use of parallel operations and falling-back may result in additional requirements upon the *prior* system that were not included in the original design.

11.3.7 Software Transition to Sustainment

Early in the program acquisition planning, the government must determine how the software in the acquired system will be sustained. The options are: the developing contractor maintains the software, the government maintains the software, or the government selects a contractor other than the developing contractor to maintain the software. The Aerospace Report No. TOR-2013-00693, *Software Sustainment Guidance* [15], provides a more complete discussion of software sustainment. Every deployment of software to operations necessitates subsequent software maintenance.

The software development contractor should have a plan for how software will be transitioned into sustainment, for both internal and external deployments. These transitions may occur multiple times during the software development lifecycle. If the contractor is using an iterative software development lifecycle model, the MSBP should identify, for each iteration, what software, if any, needs to be subsequently maintained.

Preparing for software maintenance in each iteration includes: preparing the executable software; preparing source files; preparing version descriptions for the maintenance site; preparing the "as-built" software architecture, design, and related information; updating the system/subsystem design description; updating the software requirements; updating the system requirements; preparing maintenance manuals (computer programming manuals and firmware support manuals); and transitioning the software to the designated maintenance site(s). See the SDSMCS [14] for more detail about the requirements for software transition to sustainment.

11.4 Software Acquisition Best Practices

This section provides the software acquisition best practices currently identified as being significant contributors to project success and are based upon inputs from people with recognized expertise in software acquisition. These best practices are derived from:

- FFRDC software acquisition research team experience, with many years of experience in software acquisition, software development, software management and software process improvement.
- Software Program Manager's Network, The Program Manager's Guide to Software Acquisition Best Practices, Version 2.31 [23]

To be considered best practices these practices had to be suitable for acquiring today's large, complex, software-intensive systems and for acquiring complex software systems being developed with the latest software development process and product technologies. The collection of practices have to cover the full acquisition and development lifecycle, including both pre- and post-contract award. They must be consistent and address this integrated approach throughout the acquisition, and development lifecycle. The best practices are a part of an integrated approach to system acquisition acknowledging that software always exists within the context of the system. Finally, these best practices must be consistent with applicable system and software acquisition policies and regulations.

The Best Practices are provided in Table 11-1. This table contains columns that provide a mapping to the topics found in this section as follows:

Section title heading	Section & Column
Software in the Early Acquisition Lifecycle	6.2.1 Early Acquisition
Software topics in Contracting	6.2.2 Contracting
Software Development Lifecycle	6.2.3 SW Development
Software Support Activities	6.2.4 SW Support
Software Reviews	6.2.5 SW Reviews
Software Transition to Operations	6.2.6 SW Transition to operations (Ops)
Software Transition to Sustainment	6.2.7 SW Transition to Sustainment

Note: Within each of these sections, the order of the best practices does not imply any order for implementation.

Table 11-1. Software Acquisition Best Practices

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Integrate software into systems acquisition	Proactively integrate software acquisition with the system acquisition process. Ensure software is appropriately integrated from materiel development decision through system retirement, and especially during early lifecycle and pre-contract award activities	x	x	x	x	x	x	x

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Perform software acquisition risk management	Perform continuous software acquisition risk management [23] Across the entire acquisition lifecycle, across all increments, and within each ongoing increment Carefully assess and manage reuse risks and costs, include all categories of reuse (e.g., product line, government off-the-shelf (GOTS), open source software) Program-level risk management and contractor-development risk management are necessary, but not sufficient	x	x	x	x	x	x	x
Establish a Work Breakdown Structure (WBS) suitable for software intensive systems	A software systems engineering and integration WBS element should be defined at Level 2 of the WBS for cross-product software tasks, when appropriate. [18]	x	x	x	x	x	x	x
Establish an organizational structure suitable for software intensive systems	Software needs to be at an organizational level commensurate with its risk A software organization reporting to the program manager, at the same level as systems engineering, has been shown to reduce risk	x	x	x	x	x	x	x
Staff the APO with qualified software personnel	Incorporate experienced government, FFRDC, SE&I, and SETA software personnel into the acquisition program office (APO).	x	x	x	x	x	x	x

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Establish and maintain the software acquisition management plan	Plan the software effort for the entire acquisition lifecycle Perform detailed planning for current phase and near-term activities; update plans for later activities and phases as you proceed Document your plan for software acquisition management either as a part of the Systems Engineering Plan (SEP) or as a stand-alone document, sometimes called a Software Acquisition Management Plan (SWAMP) Ensure all software acquisition personnel know and follow the plan	x	x	x	x	x	x	x
Establish and maintain the APO's SW acquisition processes	Define processes for all software acquisition activities, including the support processes (e.g., configuration management (CM), SW acquisition risk management, etc.) Document the processes; update as needed Ensure all software acquisition personnel know and follow the processes Have a software acquisition process improvement plan	x	x	x	x	x	x	x

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Establish and maintain acquisition life-cycle configuration management	Adopt lifecycle CM across the entire software acquisition life cycle for the control of acquirer and contractor information. Examples of acquisition items to be under CM: Solicitation packages (e.g., RFP), government plans (e.g., risk management plan, program management plan, program protection plan, systems engineering plan, software acquisition plan), acquisition strategies, contractor deliverables, demonstration/ simulation results	x	x	x	x	x	x	x
Initial System Definition: Ensure required capabilities appropriately include software	Iterate with user to ensure software-related capabilities are appropriately included in the Interface Control Document (ICD) and draft CDD. Examples: Key performance parameters (KPPs), reliability/ maintainability/availability (RMA), fault management, integrated HW-SW diagnostics, supportability, safety, security (information assurance, cyber security), human systems integration (HSI), interoperability, and net-centricity architecture views	x	x					
Initial System Definition: Ensure CONOPS appropriately includes software	Iterate with user to ensure the user's concept of operations (CONOPS) is feasible from a software perspective and is consistent with the required capabilities The CONOPS is embodied in the software	x	x					

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Develop a software-inclusive Test & Evaluation (T&E) strategy	For development testing, mandate compliance with standards that require a robust software development test program *Software Development Standard for Mission Critical Systems*, [14] *Test Requirements for Launch, Upper Stage, and Space Vehicles* [24] Ensure software testing follows "Test Like You Fly" principles Document as part of the program's T&E strategy	x	x					
Government reference architecture: Perform software-inclusive architecture trade studies	With system architecture trades: Identify and address critical HW/SW architecture issues Include major legacy components and COTS SW	x	x					
Government reference architecture: Include software in evaluation of architecture concepts	Evaluate software evolution and growth capability. Include software in lifecycle cost analysis (COTS software refresh, legacy and new software re-engineering and maintenance)	x	x	x				
Government reference architecture: Select a set of integrated HW-SW architecture concepts	Ensure the concepts are: Able to grow with each successive increment with little expected rework. Able to integrate each successive increment with previous increments (and legacy system, as applicable).	x	x	x				

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Determine realistic SW size estimates for each increment	Use Gov't. HW/SW architecture concept. Include all SW functionality and infrastructure needed. Use historical data from similar past programs and early concept study data.	x						
Determine realistic SW effort & cost estimates for each increment	Include COTS, reuse and new software. Include tasks not reflected in cost models (e.g., COTS refresh, integration of SW and HW items). Estimate at least at 80% likelihood level.	x						
Determine realistic SW schedule estimates for each increment	Include all software effort in schedule. Identify and include critical dependency impacts on schedule. Never compress software schedule >20% off nominal. [23]	x						

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Executable Program Increments: Consider SW implications when defining increment capabilities	Analyze feasibility of developing the required software for each increment Based on realistic software size, effort, cost, schedule estimates Include software cost and schedule estimation risk Analyze feasibility of integrating the software in each increment with all previous increments (and legacy system(s), as applicable) Based on integrated hardware/software architecture Analyze feasibility of developing required software in EMD–P&D phases vs. beginning developing initial builds in TMRR Phase (for each increment).	x						
Executable Program Increments: Consider SW implications when defining increment schedules	Analyze feasibility of overlapping software development schedules for closely spaced increments. Avoid plans that require developing subsequent increments on unknown software baselines. Analyze feasibility of COTS refresh and legacy SW upgrades schedules.	x	x					

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Software risk assessment: perform a software technology readiness assessment (TRA)	Include software in the required pre-Technology Readiness Assessment (TRA), including: COTS (e.g., operating system, development environment) Software product technologies Software process technologies Ensure technologies related to software are appropriately included in the TRA (e.g., computer system hardware, new algorithms). Identify software critical technology elements (CTE). Include software and software-related technology maturation in technology development strategy for the TMRR Phase.	x	x					
Software risk assessment: perform a software risk assessment	Identify and assess software risks for the entire acquisition lifecycle as part of the program risk assessment. Include software technical risks and software process risks as well as software cost and schedule risks.	x	x					
Software risk reduction in the MSA phase	Include efforts to reduce software risk and prepare for material solution analysis (MSA,) e.g., • Include software in engineering analyses of alternative solutions. • Perform software architecture studies as part of system architecture studies. • Estimate software size, cost and schedule. • Identify software CTEs and develop strategies for technology maturation.	x						

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Acquisition Strategy: Develop plans for computer system technology insertion	Include COTS HW and SW refresh in each successive increment. Understand new computer HW and SW technologies needed for each increment and assess their readiness.	x	x	x	x	x	x	x
Acquisition Strategy: Develop plans for software support	Plan for managing multiple baselines (operations and development). Plan for integrating software maintenance actions on operational increments into increments under development. Plan for obtaining needed data and data rights.	x	x	x	x	x	x	x
Acquisition Strategy: Develop plans for evaluation of contractor software capability	Establish plans to perform a government evaluation of contractor team software capability. Tailored evaluation as part of source selection for TMRR end-item design contract. During TMRR or part of source selection for the single development contractor.	x	x					
Software Acquisition Management Plans: Ensure a software-inclusive systems engineering plan (SEP) is developed for the program	Ensure software systems engineers are included in the preparation and updates of the SEP throughout the acquisition lifecycle, so that it is a full systems engineering plan, and not just a hardware systems engineering plan.	x	x	x	x	x	x	x

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Software Acquisition Management Plans: Develop and maintain the plan to manage software acquisition activities	Develop and update when needed, the plan to manage software acquisition for the program. This may be a plan included as an attachment to the SEP, or a separate plan such as a software acquisition management plan (SWAMP). Ensure the SWAMP and SEP are consistent with each other and with other program plans (e.g., the technology development strategy, the test and evaluation strategy, the acquisition strategy).	x	x	x	x	x	x	x
Contract for software technology risk reduction	Include efforts to reduce software risk in TMRR and competitive prototyping contracts, e.g., Software technology maturation. Software prototyping for high-risk areas of the software (e.g., difficult KPPs such as timing requirements, complex mission planning concepts, new algorithms, software architecture foundation). Prototyping of new software process technology. Prototyping candidate COTS software products.	x	x					

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Developing Test and Evaluation Plans: Develop SW-inclusive test and evaluation plans	Ensure software is effectively included in: Measures of operational effectiveness and suitability Critical technical parameters, including software maturity and technical performance measures (TPMs) Ensure robust plans for software are integrated into the program plans for both developmental test and evaluation (DT&E) and operational T&E (OT&E) Use robust software test standards for DT&E [14] Ensure DT&E and OT&E software testing follows test-like-you-fly principles Document in the program's test and evaluation master plan (TEMP) [25]		x					

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Establishing the requirements baseline: Include software in government. system performance requirements	Include software in government system performance requirements: KPPs Requirements derived from CONOPS Specialty engineering, especially RMA, safety, security (information assurance), HSI Supportability, fault management, integrated HW/SW diagnostics, COTS and OSS software supportability Open system architecture, interoperability, standardized interfaces Net-centricity Computer resource margins and growth; software evolvability Document in system requirements document (SRD) or TRD, the contractual requirements document	x	x					
Establishing the requirements baseline: Contract for delivery of SW-inclusive requirements specifications and SRR	Require System or Segment Specifications (or both) as CDRL items. Use System/Subsystem Specification Data Item Description (DID) (DI-IPSC-81431A) for content requirements Contract for a software-inclusive SRR		x					

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Contract for studies to support development of the system architectural design	Contract for software-inclusive system architecture studies: • Include software as well as hardware in the system architecture definition • Include major software legacy components and COTS software • Contract for delivery of SW-inclusive system architecture and SDR • Require system architecture as a CDRL item • Require an integrated HW/SW architecture, including: • DOD Architecture Framework (DoDAF) views • Top-level computer system/software architecture • Newly developed, reuse, OSS and COTS software • Use tailored system/subsystem Design Description DID (DI-IPSC-81432A) • Contract for a software-inclusive SDR		x					

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Developing the System/Segment Integration and Verification Plan (SSIVP): Contract for software-inclusive system test planning	Contract for top-level software-inclusive development test planning that addresses all levels and types of development testing for TMRR, EMD–P&D phases: • Software, hardware, spacecraft subsystem, spacecraft, payload, space segment, ground element, ground system/segment, system. • Software unit, integration and qualification testing. • Ensure plans comply with robust software testing standards and "test like you fly" principles. [14]		x					
Developing the SSIVP: Contract for delivery of software-inclusive system test planning	Require a software-inclusive SSIVP as a CDRL item (called by many different names, e.g., master test plan, test and evaluation plan). Use the technical studies miscellaneous DID (DI-MISC-80805A) with contents specified. Provides input for the TEMP.		x					
Software Process Risk Reduction: Contract for definition of full lifecycle software processes	Require delivery of software development plan (DID DI-IPSC-81427A—Tailored for the program). Require the SDP to address the full software development lifecycle, including TMRR, EMD, and P&D phases and implementation. SDP prepared and provided during proposal phase, delivered as a CDRL as a part of engineering development and production, and then updated as needed. Identify SDP as a compliance document.		x					

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Software Process Risk Reduction: Contract for prototyping of new software processes	Require prototyping of any new software development processes, methodologies, tools or techniques. Require their application on SW development to be done in TMRR phase (e.g., prototypes, simulators).		x					
Software Process Risk Reduction: Assess the software and systems engineering processes	Contract for a software process appraisals of the software and systems engineering processes proposed for the full development lifecycle (TMRR and EMD phases), covering all software team members. Contract for periodic process appraisals on approximately 18 month cycles Either done by the government team or by the contractor with government participation and an independent certified lead. Use software experts trained in process appraisals to lead or be on team. Require a periodic delivery of a CDRL documenting a process improvement plan based on the appraisal results (use a miscellaneous DID).	x	x					

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Contract to obtain software process contractual commitment	Ensure RFP and contract require: • Compliance with SDP in SOW. • Mandated compliance with robust full lifecycle and software development standards • "SW Development Standard for Space Systems", tailored for the program. [14] • Require other software-related standards that complement the full lifecycle standard (e.g., ISO 15939 for metrics, IEEE 1471 for software architecture descriptions). • SDP to be compliant with the standards for the full lifecycle (TMRR, EMD, P&D and O&S phases) and all contract provisions. • Require access to SDP-referenced processes and procedures. • Require contractor commitment to Software Development Plan. • Include commitment to SDP in IMP for the work done for each program phase.		x	x	x	x	x	x

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Obtaining visibility into software products: Contract for key software technical and management deliverables	Recommended contracted software CDRL items, specifying electronic delivery: • Software development plan • Software master build plan • Software transition plan • Software requirements and interface requirements specifications • Software architecture description • Software test plans, procedures and reports • Software and system metrics reports • Software and system change records • Delivery, installation and maintenance documentation • Access to subcontractor developed products		x					
Obtaining Visibility into Software Products: Contract for software level technical and management reviews	Require software level technical and management reviews. • Contract for software-inclusive PDR and TRRs. • Contract for software increment (build)-level reviews. • Require access to contractor's review of subcontractors products		x	x	x	x		

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Obtaining Visibility into Software Work Products: Contract for timely electronic access to all software products.	Require timely electronic access (e.g., DAL) to all software work products, such as: • Intermediate and final products • Requirements, architecture, design • Requirements traceability tool(s) • Implementation (including code) • Integration and verification testing • Trade studies, engineering memos • Management information, change control records, metrics.		x	x	x	x		
Software Product Risk Reduction: Contract for use of techniques to "prove" the software architecture.	Contract for use of dynamic modeling and simulation and engineering analyses to demonstrate that the software architecture will work and meet critical software performance requirements. Extend details of the TMRR modeling and simulation with detailed design and measurements of code execution.		x	x	x	x		

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Software Product Risk Reduction: Contract for use of prototyping of software high risk areas.	Require prototyping of high risk areas of the software through the design phase: • Difficult KPPs • Complex mission planning concepts • New algorithms • Difficult timing requirements • Critical interfaces • SW architecture foundation and design foundation • Continuation of earlier competitive prototyping efforts to refine the software requirements and architectural design.		x					

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Proactive Software Contract Monitoring: Contract and use award/incentive fee to reward high quality SW products and processes aggressively	Incentivize software quality*, not just cost and schedule. Use award/ incentive plans, reward high quality SW products & processes to: • Reward adherence to • Defined software processes • Software process improvement • Reward timely and adequate response to government comments • Reward "proven" SW architectural designs • Reward ground SW architecture and implementation that supports COTS SW evolution and legacy transition • Reward effective SW peer reviews and SW test program • Reward meeting RMA requirements during software and system testing and post-delivery & launch • Reward high quality SW products • Reward successful prototyping of high risk areas **Quality in this context is producing work products that do not require rework in successor activities.*		x	x	x	x	x	x

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Prepare RFPs and monitor contracts: Refine and update the products generated as a part of earlier phases:	Prepare RFPs and monitor contracts							
	Update the requirements baseline							
	Include software in update of government system performance requirements (updated SRD or TRD for EMD–P&D contract)							
	Contract for delivery of software-inclusive requirements specifications (updated system and segment specs)							
	Update the system architectural design							
	Contract for delivery of software-inclusive system architecture (updated system architecture description)		x					
	Update the system/segment integration and verification plan							
	Update Government acquisition documents [e.g., acquisition strategy plan (ASP,) Capability description document (CDD), cost analysis and requirements document (CARD), SEP, SWAMP]							
	Contract for delivery of software-inclusive system test planning (updated SSIVP)							

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Selecting a Capable Software Contractor Team: Evaluate software capability as part of source selection.	Require an SDP covering all software team members to be provided with the proposal. If the government had not performed a process appraisal during the TMRR phase, perform a process appraisal of the software team members' software and systems engineering processes proposed for this program during source selection. If the government had performed a process appraisal during the TMRR phase, require the process improvement plan and results to date to be provided with the proposal.		x					
Selecting a Capable Software Contractor Team: Evaluate software architecture with system preliminary design.	Evaluate major HW/SW architecture issues (e.g., space-ground trades, use of COTS, OSS and legacy components, proposed software architecture)		x					
Selecting a Capable Software Contractor Team: Evaluate real cost and schedule bids.	Evaluate for extremes in cost and schedule in productivity, COTS, OSS and reuse, and low lines of code Ensure all COTS and OSS SW tasks included Ensure bids contain enough margin.		x					

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Performing Independent Validations: Perform independent evaluation of SW architecture definition and design	Independent evaluations of the contractor team's software architecture and design. Independent modeling and simulation of the software architecture and design to verify that the architecture and design will meet the performance requirements Extend details of TMRR modeling and simulation with detailed design and measurements of code execution.		x	x		x		
Performing Independent Validations: Perform independent evaluation of SW cost and schedule plans	Perform: Independent evaluation of the contractor team's software size estimates. Independent evaluation of the contractor team's software cost and schedule estimates using static models (e.g., cost and schedule models). Independent evaluation of the contractor team's planned software increments (builds) using dynamic modeling and simulation (e.g., system dynamics modeling, discrete event simulation).		x	x		x		
Proactive Software Contract Monitoring: Perform in-depth technical reviews of software products and processes.	Use software experts for in-depth SW product and process reviews. Perform in-depth SW product reviews: • SW CDRL item reviews. • Design and software build reviews. • Review intermediate and final versions of key SW products,		x	x	x	x	x	x

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
	whether or not they are deliverable. • High risk, critical software products. • Key software technical deliverables Participate in: • SW and system architecture, requirements and design reviews. • IPTs, Technical Interchange Meetings (TIMs), working groups, peer reviews, etc. • SW increment (build)-level reviews and SW-inclusive major reviews • Monitor SW testing (e.g., for increments (builds)) Perform in-depth SW process reviews: • Review program team's adherence to their defined software processes • Identify adherence deficiencies • Assist in deficiency correction • Evaluate effectiveness of their defined processes (e.g., they are providing value to the program by reducing defects, cost or schedule) • Identify process deficiencies (to improve value) • Assist with process improvement							

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Proactive Software Contract Monitoring: Perform independent gov't. SW assessments and analyses.	Perform independent government SW assessments and analyses: • Independent software architecture assessments • Independent modeling and simulation of software architecture	x	x	x	x	x		
Proactive Software Contract Monitoring: Apply proactive quantitative management	Ensure a comprehensive software and system metrics program, balanced across information categories. Include leading quality indicators Perform independent analysis of contractor metrics, including cross-metric analysis and trending. Take management action based on metrics analysis.		x	x	x	x	x	x
Performing Software Product Reviews: Include users & operators in SW technical review activities.	Focus on operational suitability of evolving software-intensive system. Including COTS software capabilities and impacts on O&S.		x	x	x	x	x	x

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Performing Software Product Reviews: Monitor software integration and verification adequacy.	Begin at the SW increment (build)-level. Focus on areas of highest risk. Focus on early performance analysis results and meeting KPPs. Ensure performance is measured, including COTS SW performance. Ensure all SW requirements are verified, including those met by COTS.	x	x	x	x	x		
Performing Software Process Reviews: Review effectiveness of contractor team's SW processes.	Review team's adherence to their defined software processes Identify adherence deficiencies Assist in deficiency correction Evaluate effectiveness of their defined SW processes Identify process deficiencies Assist with process improvement A high maturity level for the prime contractor or any team member will not ensure the processes in use are effective.	x	x			x	x	x
Performing Software Process Reviews: Perform periodic team software capability process appraisals	Perform during contract period of performance Just because the contractor says their organization is a certain maturity level, it doesn't guarantee those processes are being used on your program. Track progress against process improvement plan Use improvement results in award fee determination	x	x					

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Perform Software Maintenance Actions	For production contracts, apply the same SW acquisition best practices as the TMRR and EMD–P&D development contracts. • Production usually includes some software requirements changes • These changes can be major software upgrades • Production will always require software maintenance actions (e.g., COTS and OSS upgrades, bug fixes) • For software, each maintenance action is a mini-development action		x				x	x
Operations and Sustainment Planning	Successful sustainment requires planning from the beginning (MSA Phase) • Responsible organization • Documentation, data rights • Funding and schedule for software support, including COTS and OSS software refresh • Managing multiple baselines	x	x				x	x
Apply the same best practices	Apply the same software acquisition best practices as for TMRR and EMD–P&D Phases • For software, each maintenance action is a mini-development action • Some sustainment contracts include major software upgrades		x	x	x	x	x	x

Best Practice	Details	6.2.1 Early Acquisition	6.2.2 Contracting	6.2.3 SW Development	6.2.4 SW Support	6.2.5 SW Reviews	6.2.6 SW Transition to Ops	6.2.7 SW Transition to Sustainment
Define appropriate software metrics	Identify the government information needs for sustainment and define metrics accordingly • Need additional metrics related to repair		x		x		x	x
Provide incentives	Incentivize software maintenance quality (e.g., low rework rate on modified code) and responsiveness to operational needs		x				x	x

11.5 References

1. Gallagher, B. P., M. Phillips, K. Richter, S. Shrum. *CMMI® for Acquisition; Guidelines for Improving the Acquisition of Products and Services*, Version 1.3, Second Edition, Addison-Wesley. 2011.

2. Chrissis, Mary Beth, M. Konrad, and S. Shrum. *CMMI® for Development; Guidelines for Process Integration and Product Improvement*, Version 1.3, Third Edition, Addison-Wesley. 2011.

3. Systems and software engineering —Software life cycle processes, ISO/IEC/IEEE 12207:2008, Second Edition, 1 Feb 2008.

4. Systems and software engineering – Vocabulary, ISO/IEC/IEEE 24765, First edition, December 15, 2010.

5. Abelson, L. A., R. J. Adams, A. B. Arehart, M. J. Hecht, L. J. Holloway, D. J. Naiditch, and R. M. Wilkes, *Integrating Software Topics Into the Request for Proposal*, TOR-2011(8506)-117. The Aerospace Corporation, El Segundo, CA. July 19, 2012.

6. Sather, V., et al., *General Guidance on Open Source Software*, TOR-2014-02681. The Aerospace Corporation, El Segundo, CA. 2014.

7. Defense Federal Acquisition Regulation Supplement, Acquisition Regulations, Washington, D. C., DOD, 1998. See htthp://farsite.hill.af.mil, accessed March 2012.

8. Clements, P., R. Kazman, and M. Klein, *Evaluating Software Architectures: Methods and Case Studies*, Addison-Wesley. 2002.

9. Unell, A., et al., *Evaluating Software Architectures in Space and Ground Systems*, ATR-2012(9010)-12, The Aerospace Corporation, El Segundo, CA. 2012.

10. Paulk, M., et al, *The Capability Maturity Model for Software: Guidelines for Improving the Software Process*, Addison-Wesley. 1994.

11. CMMI® Insititute-2013-HB-001, *Standard CMMI® Appraisal Method for Process Improvement (SCAMPI SM) Version 1.3a: Method Definition Document for SCAMPI A, B, and C,* October 2013.

12. Owens, K. L., and J. M. Tagami, *Recommended Software-Related Contract Deliverables for National Security Space System Programs*, TOR-2006(8506)-5738, The Aerospace Corporation, El Segundo, CA. 2008.

13. Owens, K. L., and J. M. Tagami, *Recommended Software-Related Systems Engineering Contract Deliverables for National Security Space System Programs*, TOR-2005(8506)-8101, The Aerospace Corporation, El Segundo, CA. 2008.

14. Adams, R. J., et.al*., Software Development Standard for Mission Critical Systems*, TR-RS-2015-00012, The Aerospace Corporation, El Segundo, CA. March 17, 2014.

15. Adams, R. J. and S. Eslinger, *Software Sustainment Guidance*, TOR-2013-00693, The Aerospace Corporation, El Segundo, CA. December 31, 2013.

16. Abelson, L. A., et al., *Software Measurement Standard for Space Systems*, TOR-2009(8506)-6, The Aerospace Corporation, El Segundo, CA. May 5, 2011.

17. Peresztegy, L. B. (Sam) and C. E. O'Connor, *Technical Reviews and Audits for Systems, Equipment, and Computer Software*, TOR-2007(8583)-6414, Volume 1, The Aerospace Corporation, El Segundo, CA. January 30, 2009.

18. Peresztegy, L. B. (Sam) and C. E. O'Connor, *Technical Reviews and Audits for Systems, Equipment, and Computer Software: Space Systems Supplement*, TOR-2007(8583)-6414, Volume 2, The Aerospace Corporation, El Segundo, CA. September 30, 2009.

19. MIL-HDBK-881, *Work Breakdown Structures for Defense Materiel Items,* Department of Defense Handbook, Washington D. C., DOD. 2005.

20. Eslinger, S., *The Position of Software in the Work Breakdown Structure (WBS) for Space Systems*, TOR-2006(8506)-5738, The Aerospace Corporation, El Segundo, CA. 2006.

21. Holloway, L. J., et al., *Iterative Software Development in Space Systems Acquisition,* TOR-2013-00692, The Aerospace Corporation, El Segundo, CA. June 1, 2014.

22. Eslinger, S., L. J. Holloway, and R. M. Wilkes, *Space Segment Software Readiness Assessment*, TOR-2011(8591)-20, The Aerospace Corporation, El Segundo, CA. June 3, 2011.

23. Software Program Manager's Network, *The Program Manager's Guide to Software Acquisition Best Practices*, Version 2.31,

24. Perl, E., *Test Requirements for Launch, Upper-Stage, and Space Vehicles*, TR-2004(8583)-1 Rev. A, The Aerospace Corporation, El Segundo, CA.

25. *Defense Acquisition University (DAU) Lessons Learned, Proven Practices* portal, https://acc.dau.mil/CommunityBrowser.aspx?id=639146&lang=en-US

11.6 Acronyms

APO	Acquisition Program Office
ASP	acquisition strategy plan
ATP	authority to proceed
CARD	cost analysis and requirements document
CDD	capability description document
CDR	critical design review
CDRL	contract data requirements list
CM	configuration management
CMMI®	Capability Maturity Model Integration
CMMI®-DEV	CMMI® for Development
CONOPS	concept of operations
COTS	commercial off-the-shelf
CTE	critical technology element

DAL	data accession list
DAU	Defense Acquisition University
DFARS	Defense Acquisition Regulations Supplement
DID	data item description
DoDAF	DOD architecture framework
DT&E	developmental test and evaluation
EMD	engineering and manufacturing development
ETVX	entry, task, verification, exit
FFRDC	Federally Funded Research and Development Center
GFP	government furnished property
GOTS	government off-the-shelf
HSI	human systems integration
HW	hardware
IBR	integrated baseline review
ICD	interface control document
IEEE	Institute for Electrical and Electronics Engineers
IMP	integrated master plan
IPT	integrated product team
ISO	International Organization for Standardization
KPP	key performance parameter
MSA	materiel solution analysis
MSBP	master software build plan
NSS	national security space
O&S	operations and support
OPs	operations
OT&E	operational test and evaluation
OSS	open source software
PDR	preliminary design review
P&D	production and deployment
RFP	request for proposal
RMA	reliability/maintainability/availability
SAR	software requirements and architecture review
SCAMPI[SM]	Standard CMMI® Appraisal Method for Process Improvement
SCM	software configuration management
SDP	software development plan
SDR	system design review
SFR	system function review
SDSMCS	software development standard for mission critical systems
SE&I	systems engineering and integration
SEP	systems engineering plan
SETA	systems engineering and technical assistance
SFR	system functional review
SOW	statement of work
SPMN	Software Program Manager's Network
SRD	systems requirements document

SRR	system readiness review
SSIVP	system/segment integration and verification plan
SW	software
SWAMP	software acquisition management plan
T&E	test and evaluation
TEMP	test and evaluation master plan
TIM	technical interchange meeting
TMRR	technology maturation and risk reduction
TOR	Technical Operating Report
TPM	technical performance measure
TRA	technology readiness assessment
TRD	technical requirements document
TRR	test readiness review
WBS	work breakdown structure

Chapter 12
Ground Segment Hardware

Randall M. Onishi
Advanced Demonstrations
MILSATCOM Division

12.1 Introduction

The ground segment consists of the Ground Station and Terminals, Mobile Ground Element, and the Mission Control Center (MCC) that includes Mission Management, Mission Data Processing and Distribution, Space Ground/Asset Command & Control, and Additional Functions (Figure 12-1). An infrastructure services layer and network layer, both of which requires hardware to implement, provide communications between all of the MCC components. Development hardware for the ground segment is required for all contractors' factories working the program. Ultimately the operational suite of hardware will be delivered and installed at the operational site. This includes work stations, displays, video matrix switches, servers, data storage, front end processing equipment, ground terminals, ground terminal support hardware, network (routers, switches, and firewalls) and crypto equipment.

12.2 Hardware Descriptions

Table 12-1 lists the needed ground segment hardware a typical program for the development and operations of the ground system. External organizations to the ground segment such as the user segment, tasking authority, and the space vehicle factory will require, at a minimum, firewalls to communicate with the ground segment.

12.3 Ground Segment Hardware Overview

Program requirements and concepts of operations will drive the design of the ground segment and the types and amount of hardware that will be needed for development and operations. Years ago programs may have had to procure special purpose ground processing hardware specifically built for their program in order to meet collection-to-user timeliness requirements. This was very costly and injected risk into the program because the development of special built hardware paralleled the development of the program. There were no assurances that the hardware would meet the requirements until it was delivered and tested.

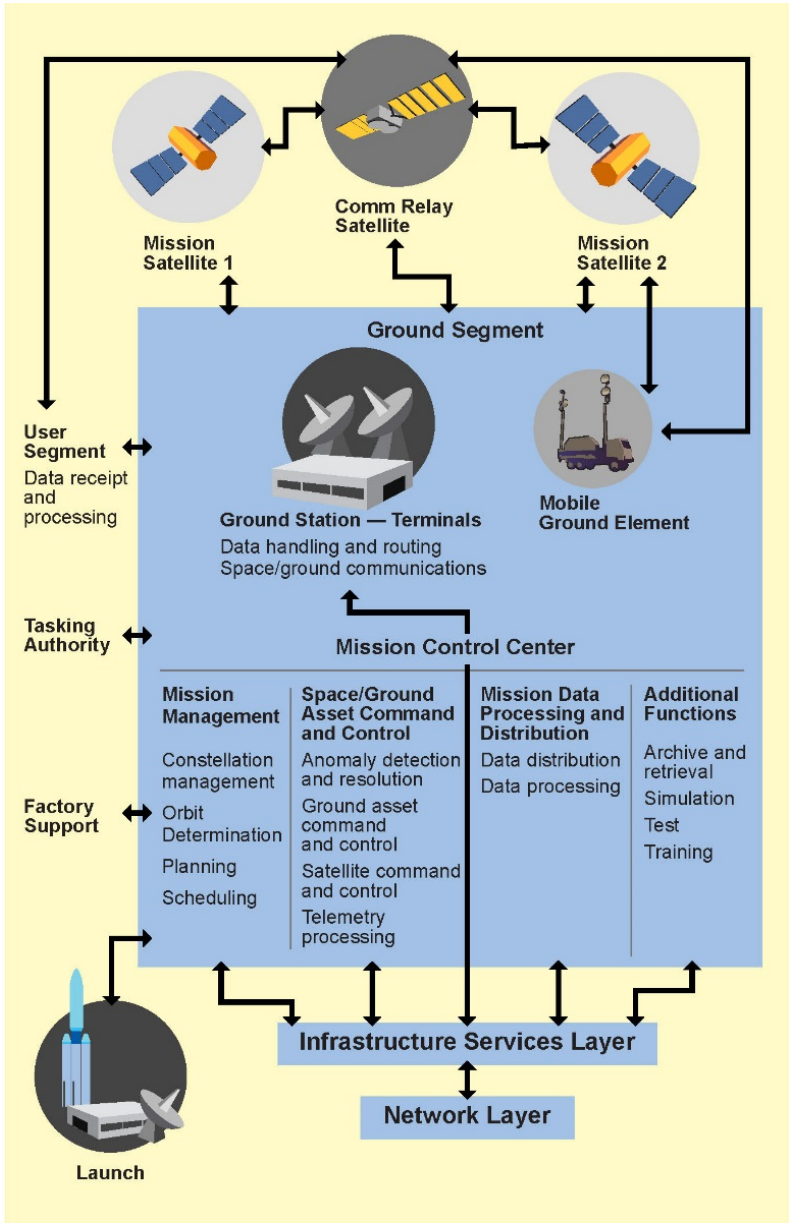


Figure 12-1. Ground Segment Components

Table 12-1. Ground Segment Hardware

Hardware	Description
PCs	Used for email, Office (Word®, Excel®, PowerPoint®), and administrative uses
Work stations	Operator consoles, development work stations, analysis workstations
Printers	Black and white or color printers. Usually hooked up to the network.
Routers	Used to connect networks and firewalls
Switches	Used to create networks and firewalls
Server	Used for firewalls and hosts small to medium applications and databases
Large server	Used to host larger applications (e.g., mission data processing) and databases
Data storage	Disks for storage of databases, mission data, telemetry, command databases, logs. Used for short term storage.
Tape storage	Long term storage media. Data is moved from short term storage to the Tape Library for archiving
Crypto equipment	Crypto equipment use to encrypt and decrypt data. Program security requirements will dictate the type of crypto equipment to acquire.
Video matrix switches	Used to control what is displayed on large screen monitors in the operations room or the anomaly resolution room
Displays	Displays for workstations and large screen displays hung on walls in operations rooms
Front end processing	Hardware used to format data for transmitting to the space vehicle and process data that is received from the space vehicle
Ground terminal	A fixed or mobile antenna used to communicate with the space vehicle

The majority of programs today can now meet the processing and timeliness requirements with general purpose commercial off-the-shelf (COTS) hardware. There may be occasions where flexible programmable gate arrays (FPGAs) may be required. Hardware for the ground segment used to be a major cost driver for programs, but that is no longer the case.

The amount of hardware required to operate the system drives the facility program requirements. The computer room needs to be sized to accommodate the hardware and meet environmental, power, security, and safety requirements. Advances in hardware design and the utilization of the hardware such as

virtualization has helped to reduce the total hardware footprint requirements. How the system will be operated also drives hardware selection and facility requirements. The amount of operator workstations and support workstations drive the size and layout of the operations floor.

Hardware that may be a cost factor for the program are the ground terminals at the fixed ground stations and the ground terminals for the mobile ground element. The ground terminals size will be determined based on the satellites on-board communications system, the programs selected transmission frequency and modulation, the location of the ground terminal, and the orbit of the space vehicle. The link budget analysis will calculate the size of antenna that will be needed to meet the requirements of the program. The number of ground terminals the program will need will be based on the orbit of the satellite, the amount of on-board data storage, and timeliness requirements. To reduce program costs most programs investigate the sharing of ground terminals already in operation. Trade studies should be conducted based on the costs for sharing and the risk of not getting the time of the ground terminal when needed versus the cost of developing, operating, and maintaining a dedicated program ground station and ground terminal.

12.4 Technical Considerations

This section discusses technical considerations regarding the development approach for the ground segment that can reduce the amount of hardware required. Two alternate considerations to purchasing all new hardware for a program are (1) implementing the program into a multi-mission command and control center (C2) and (2) use Cloud computing resources.

12.4.1 Multi-Mission Command and Control Centers

Multi-mission C2 centers are in operation today and are looking to add new programs. An example of a multi-mission control center is the Multi-mission satellite control center (MMSOC) ground system architecture (GSA) running at the research, development, test, and evaluation (RDT&E) support complex (RSC) at Kirtland AFB. Figure 12-2 is a diagram of the multi-mission satellite control center (MMSOC) GSA showing how the implementation shares hardware and software components for multiple missions. Commands are extracted from the operations database and passed on the local area network to an available Red Front End Processor (FEP), through a matrix switch onto an available KI-17, then to an available Black FEP, and then onto the ground terminal for unlinking to the space vehicle. The MMSOC GSA has incorporated the use of virtualization on the servers to create redundancy and reduce the amount of hardware footprint. The advantage of implementing a program into a multi-mission C2 center like the MMSOC GSA is that programs can share hardware resources and make use of common infrastructure components. Both

result in a development cost savings to programs. It will also save integration and test costs.

Another consideration that the program must address is the sharing of the operations floor with the other programs. If the concept of operations of the program cannot operate the program in this type of environment then this may not be an option for the program.

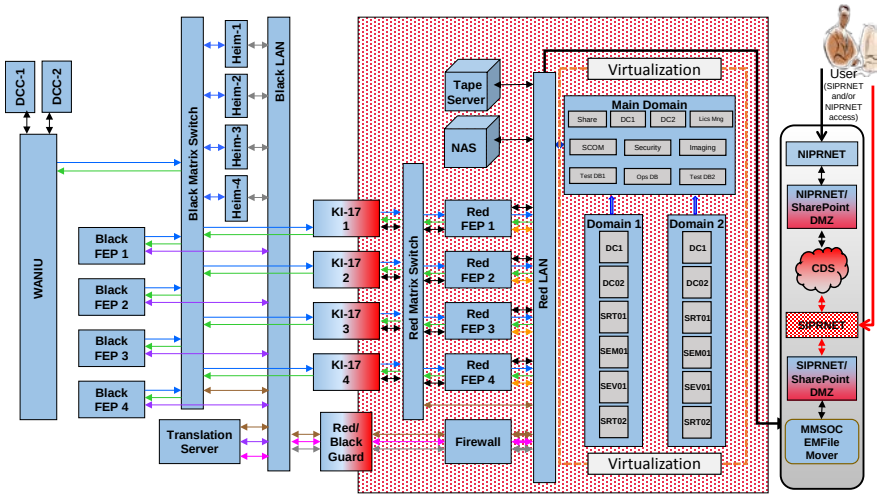


Figure 12-2. MMSOC ground system architecture.

Each multi-mission C2 center has a different way of charging programs for the shared usage of the hardware. Some may require a program to purchase hardware to increase the capacity of the center. The new hardware would not be a dedicated program resource, but a shared one. Programs will have to negotiate with the multi-mission C2 center for initial and yearly costs. Hardware maintenance costs for the hardware at the C2 center will be part of the yearly costs.

Figure 12-3 identifies the components of the ground segment MCC that the multi-mission C2 systems have implemented. The mission management and mission data processing and distribution components are not part of the multi-mission C2 center. Programs will have to independently procure hardware and have a facility to install and operate the mission management and mission data processing parts of the ground system. Traditionally the mission data processing and distribution function requires the most hardware.

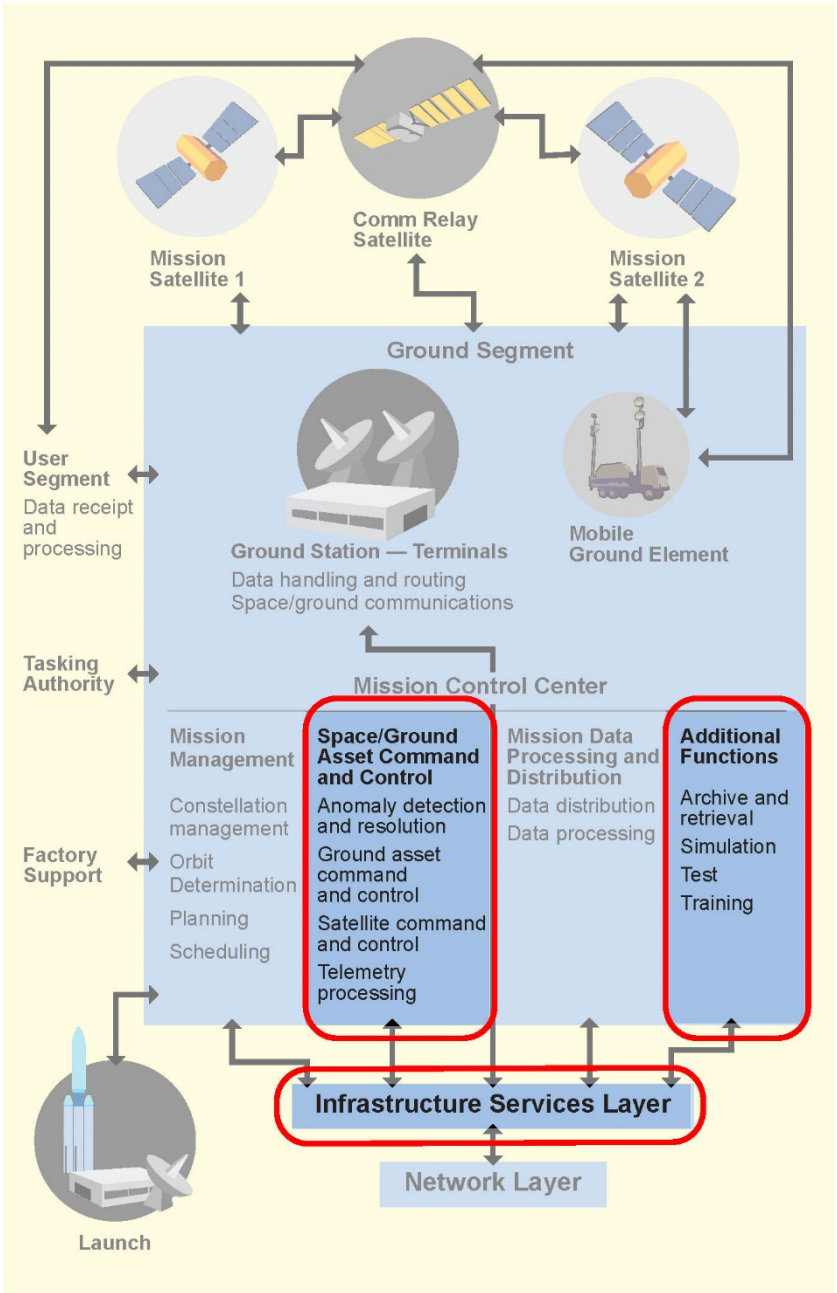


Figure 12-3. Components implemented in multi-mission C2 systems.

12.4.2 Cloud Processing

Cloud processing could be an option for the mission data processing component because it usually requires the most hardware. Other ground components can be hosted in the cloud. That would be a program decision. Considerations that need to be taken into account for cloud processing is ensuring that the network bandwidths into and out of the cloud center is adequate to handle the amount of data that will be sent for processing. In addition to the processing of the data, the archiving of the raw mission data and the processed data product will be stored in the cloud.

When considering the use of a cloud provider, negotiated service-level agreements (SLAs) will specify the services that will be provided to the program. SLAs will ensure that key cloud services (warranties, guarantees, reliability, and performance) are defined and enforceable. In addition SLAs should contain common terms and definitions to avoid misunderstandings between parties. Costs and services provided by cloud providers will vary.

12.4.2.1 Advantages of Cloud Processing

The primary advantage to cloud processing is the program does not have to procure hardware for the program for the MCC components that will use cloud processing. The cost to use cloud processing varies, and each program will still have to perform a cost/trade analysis to determine if the use of the cloud resources is cost effective.

Factors that contribute into the cost analysis are the projected length of the program and the amount of hardware required for processing and storage. Programs typically go through a hardware recapitalization every five years. The advantage of using cloud resources is the cloud center is responsible for the updating of computing resources. The costs for keeping the hardware current and maintained are costs that are passed on the users of the center.

12.4.2.2 Disadvantages of Cloud Processing

The use of the cloud for processing and data storage currently is a security concern. It is a challenge for the government to build cloud services that can be trusted and secure for multiple levels of security. Another major concern of cloud processing is what happens if the network goes down between the MCC and the cloud center. Another concern is if the network between the ground terminals that is passing raw mission data for processing to the cloud center goes down. The handling of these failure scenarios is a contactor responsibility that will require hardware to set up redundant paths. Tthese costs must be considered when investigating the use of the cloud.

12.4.3 Algorithm Scalability

Programs that have a requirement to process large amounts of data and have timeliness requirements need large servers with lots of memory. Large servers with lots of memory are expensive.

One case study made an attempt to reduce costs so contractors would purchase less expensive servers with the same hardware architecture and develop their algorithms with smaller datasets. Contractors designed and implemented the algorithms assuming they were scalable to process the larger datasets. In many cases requirements were sold off at the factory on the smaller servers using test and data analysis as the test criteria. When it came time to integrate and test the system at the operational site on the operational hardware suite, the mission data processing algorithms failed to correctly process the larger datasets. Problems that were discovered when processing the larger datasets had to do with the way the data was being handled in memory and timing issues if there was parallel processing. These problems were eventually fixed, but at a high cost because developers had to travel to the site to fix all of the problems. The strategy that the program had to save costs by not purchasing the larger server for development did not save the program money. Assuming algorithms can scale to handle larger amounts of data is not a good assumption.

This was is a case where cloud processing would be advantageous. Programs will be able to develop algorithms on the operational sized datasets and have access to hardware that will be used during operations during the development phase.

12.5 Programmatic Considerations

There are programmatic considerations that should be considered during the procurement phase of the program.

12.5.1 Backup Sites

The program will need to specify if a backup site for satellite operations is required. If so, will the backup site be a “warm” or “hot” backup site. In both cases the sites will require hardware identical to the hardware at the operational site. The backup site will need to have the space at the backup facility for both the computers and an operations floor to operate the satellite. Usually the operations floor at backup sites are identical to the layout at the primary site.

To meet the “hot” backup requirement additional hardware and data storage will be needed to ensure the primary and backup sites are “in-sync” so that if the primary site were to go offline, the backup site could immediately take over the responsibility of flying the primary site’s space assets.

12.5.2 Timing of Procurement of Operational Hardware

Usually ground system developments range from 3 to 5 years depending on the size of the program. It could be longer but 3 to 5 is the average length. With development phases being 3 to 5 years, there is always the risk that the hardware vendor comes out with an updated version of the hardware during the development phase. Programs will be faced with the challenge of whether to update to the new versions right away or wait until the next release.

This very issue changed the philosophy of when programs purchase the operational hardware. Programs now procure the hardware for the operational sites so that the hardware will be installed 18 months to a year before launch. This allows time for the software to be integrated and tested and support all of the required training and rehearsals prior to launch. It also allows time for software updates that fix problems that were not seen at the factory.

12.5.3 Hardware Maintenance Strings at Operational Sites

There should be enough hardware at the operational site to be able to upgrade software on the operational hardware without any system downtime. There is too much risk for programs to take the approach of "load-and-go" with software patches or upgrades. Some programs use this approach because they do not have enough hardware at the operational site to test the software before it is put into operations. Availability requirements limit the amount of downtime, and sometimes the program allocate time to test software. This has been a lessons learned by programs. To mitigate this issue, programs are designing an extra string, or maintenance string, of hardware that can be used to test software upgrades without any downtime.

The software fix or upgrade is loaded onto the maintenance string and tested. Once the upgrades have been verified, the system will be reconfigured to use the maintenance string as a primary processing string of hardware. The hardware string that was replaced by the maintenance string will then be upgraded and tested and then reconfigured back into operations.

12.6 Summary

With the rapid evolution of hardware technologies, programs procuring dedicated hardware run the risk of implementing ground segment capabilities on outdated hardware. Some programs may decide that it is worth the risk and cost just because they feel that having complete control over the development reduces overall program risks.

In the past there have not been many options other than to procure dedicated program hardware. There are now a couple of options that programs should

consider that might reduce the costs of ground segment hardware and their associated maintenance costs. One is to implement the C2 portion of the program into a multi-mission C2 center. The cost advantage of the multi-mission center is that all programs will share the hardware and maintenance costs. There are other costs each program will have to pay to be part of the multi-mission C2 center; each program should perform a cost and risk analysis on the impacts of implementing into a shared operations center. The second option is to implement portions of the ground segment utilizing cloud services. Again, programs will have to perform detailed cost and risk analysis to determine if using the hardware provided by the cloud will be cost effective and not increase risks.

12.7 Bibliography

Spinney, Timothy J., *Introduction to MMSOC*, The Aerospace Corporation, El Segundo, CA. TOR-2013-00150, May 31, 2013

Badger, Lee, David Bernstein, Robert Bohn, Frederic de Vaulx, Mike Hogan, Michaela Iorga, Jian Mao, John Messina, Kevin Mills, Eric Simmon, Annie Sokol, Jin Tong, Fred Whiteside and Dawn Leaf. *US Government Cloud Computing Technology Roadmap Volume I, High-Priority Requirements to Further USG Agency Cloud Computing Adoption*. NIST Special Publication 500-293. October 2014

Badger, Lee, Robert Bohn, Shilong Chu, Frederic de Vaulx, Mike Hogan, Michaela Iorga, Viktor Kauffman, Fang Liu, Jian Mao, John Messina, Kevin Mills, Eric Simmon, Annie Sokol, Jin Tong, Fred Whiteside and Dawn Leaf. *US Government Cloud Computing Technology Roadmap Volume II, Useful Information for Cloud Adopters*, NIST Special Publication 500-293. October 2014.

12.8 Acronyms

C2	command and control center
COTS	commercial off-the-shelf
DCC	distributed control center
FEP	front end processor
FPGA	field programmable gate array
GSA	ground system architecture
LAN	local area network
MCC	mission control center
MMSCC	multi-mission satellite control center
MMSOC	multi-mission space operations center
NIPRNET	nonsecure internet protocol router network
OPS	operations
RSC	RDT&E support complex

RDT&E	research, development, test, and evaluation
SCA	Service-level agreements
SIPRNET	secret internet protocol routing network
SRT	survivable relay terminal
WANIU	wide-area network interface unit

Chapter 13
Ground Segment Facilities

J. Denise Castro-Bran
Systems and Operations Assurance Department
Mission Assurance Subdivision
James A. Shneer
Space Superiority Systems Directorate
Space Support Division

13.1 Introduction/Background

Facilities and related infrastructure are an integral and critical part of the ground segment and system of systems for space systems/operations. Without the ground facilities and infrastructure, space assets could not be manufactured, tested, processed, launched nor commanded/tracked once they are in space. Facilities and infrastructure (and terminals) provide the platform that enable communications between space and ground. Space systems and operations begin and end with ground facilities and infrastructure.

Development of the ground infrastructure for support of space operations is a complex, lengthy, and expensive process. Failure to address requirements properly and adequately during the early phases of development increases the risk of delays and cost overruns during the later phases of development, and possibly compromises the system's operation. Late involvement of the ground systems infrastructure managers, subject matter experts, and/or stakeholders in the space system engineering process frequently causes delays and cost overruns.

Successful space operations are based on effective ground systems infrastructure (i.e., the facilities and the support equipment). There is a need for a systematic and timely integration of these elements into the space system engineering process.

Although there are a multitude of facilities/infrastructure that support space operations, this overview will focus primarily on those facilities, or ground stations, directly enabling the space-to-ground communications interface. This chapter will include basic requirements and the key features to consider when planning, operating, and maintaining ground segment facilities and infrastructure.

13.2 Definitions

The term **infrastructure** varies in definition depending on the industry. Military strategists use the term **infrastructure** to refer to all building and permanent installations necessary for the support of military forces, whether they are stationed in bases, being deployed, or engaged in operations [1].

The term **critical infrastructure** has been globally adopted to distinguish those infrastructure elements in the built environment that, if significantly damaged or destroyed, would cause serious disruption to the dependent system or organization. In the context of facilities and related infrastructure, the following are most commonly associated:

- Structural elements (buildings)
- Power (generation, transmission, and distribution)
- Environmental systems (heating/cooling systems, exhaust)
- Water supply (potable, waste water)
- Fuels and commodities
- Communications
- Security systems

In the context of this chapter, **infrastructure** is the critical infrastructure that supports facilities, in alignment with the military strategist definition.

Facility/Facilities A structure, or set of structures, used to house specific ground system elements or subsystems.

Ground Segment That part of a space system for which all elements reside on the ground. It is a subsystem within the overall system.

Ground Site A particular geographical location, which contains one or more ground system elements or subsystems.

Ground Station An element of a ground system the function of which is to provide a communications link between the ground segment and space segment. Usually provides both up/down communications links.

Mission Control Center (MCC) Responsible for creating the mission plan, scheduling spacecraft resources, and selecting ground resources to meet objectives. The MCC determines the spacecraft's orbit and attitude, sending predicted values to ground station for tracking acquisition. For complex spacecraft, the MCC may sometimes divide functions into two areas: (1) Spacecraft Operations Control Center (SOCC), which controls the spacecraft's

subsystem and processes its data; (2) Payload Operations Control Center (POCC), which analyzes the mission data.

Segments A major product, service, or facility of the system.

Ground or User Terminal Generally refers to a program/mission specific element that may have any or all of the following capabilities: send/receive data, data processing and display. In many cases these terms refer to mobile units deployed in the field.

Validation [of Requirements] Confirms that the requirements for a system/product are sufficiently correct, complete, and meet the stakeholders' needs and expectations.

"Did you define the requirements "right" to build the right thing?"

Validation [of System/Product] Confirms that development and verification of a system/product results in a product, service or system that meets expected outcomes based on initial requirements, specifications, and regulations.

"Are you building (designing, testing) the right thing?"

Verification Confirms that a product, service or system meets a set of design requirements, specifications, design drawings, and regulations/codes.

"Are you building (designing, testing) it right?"

Design Charrette A focused and collaborative brainstorming session held at the beginning of a project.

13.3 General Overview and Background

Buildings are deceptively complex and the roles they perform are constantly changing. They are expensive to build and maintain and must constantly adjust and respond to changing requirements that range from evolving owner and/or user needs to regulatory mandates. Facilities and related infrastructure must sometimes undergo significant modifications and upgrades to respond to these evolving requirements and to have the capability to function effectively over the course of their lifecycles. The economics related to building construction and renovation has become as complex as the buildings themselves.

All facilities have some common characteristics, but then there are specific aspects and requirements that must be considered that are determined by the function that a facility needs to serve.

13.3.1 Ground Segment Facilities

The spectrum of terrestrial facilities supporting space missions is extensive with copious challenges. The ground segment encompasses all facilities that are required for support, not just the ground segment (GS) itself, but also the other segments such as the launch segment (LS), user segment (US), and space segment (SS), and their related functions. The overview on Figure 13-1, adapted from the Aerospace Institute's ground systems tutorial course charts [2], shows the top-level interplay between all the space and ground systems. The facilities within the ground segment are critical in supporting these interconnections, as are the ground terminals and mobile/transportable system.

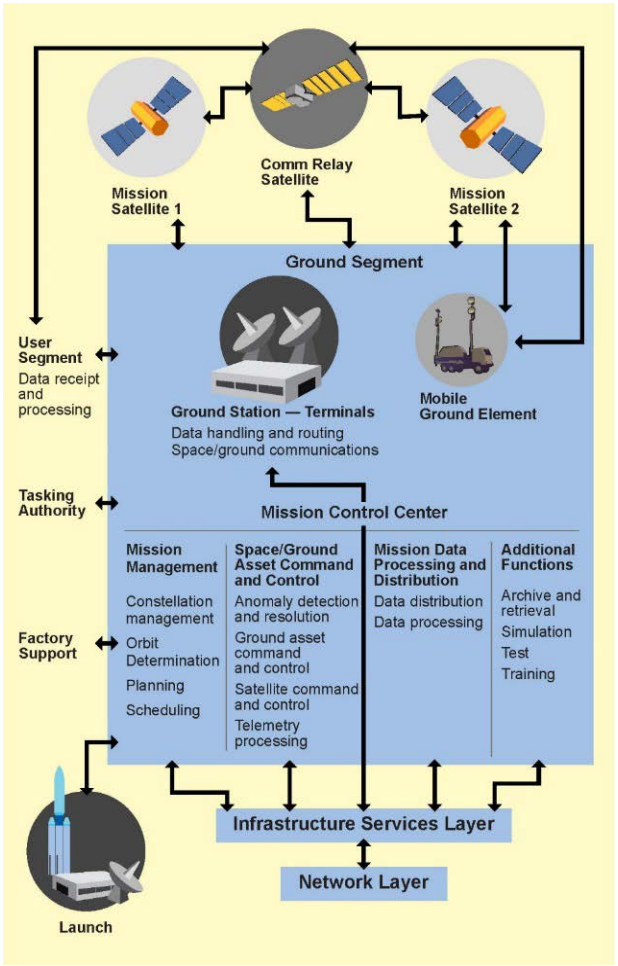


Figure 13-1. Ground segment overview.

13.3.1.1 Ground Station

The ground station is an installation on Earth that houses the equipment and functions that enable satellites and spacecraft communications. Ground stations range in size, from very small to very large and extensive. The complexity of the facility also varies, depending on the size and function of the operations the facility houses. Ground stations typically operate 24/7, given that satellites operate (orbit) continuously.

13.3.1.2 Ground Station Functional Areas (Operations)

Added to the complexity of the ground station installation, are the functions which it performs. The three primary functions that can be performed within ground stations are as follows:

- Space/Ground asset command and control (control)
- Mission management (monitoring)
- Mission data processing and distribution (analysis)

13.4 Technical Considerations

13.4.1 Site Identification and Selection

There are many considerations when selecting a site for any facility, but the requirements of locating a ground station pose further challenges.

There are several ways at looking at satellite ground facilities (refer to Figure 13-2). A fundamental categorization, shown in the top half of the figure, is based on the functions contained within the facility; there are stations that require view of the satellite(s) and those that do not. Facilities engaged in activities such as mission data processing or mission planning do not require satellite contact and are, from a facilities perspective, essentially office (administrative) buildings. Facilities that require satellite contact and are performing satellite command/control, acquisition of telemetry, and/or acquisition of mission data, are considered ground stations. This chapter discusses the characteristics and considerations for ground stations.

A second aspect of satellite ground facilities is the operating environment, as shown in the bottom half of the Figure 13-2 (land-based, sea-based, or sited aboard aircraft). This chapter addresses only the land-based facilities.

A third characteristic of satellite ground facilities relates to the degree of mobility required ranging from sites which are constructed at permanent fixed

sites through varying degrees of mobility collectively described as non-fixed sites. This chapter discusses the fixed, or permanent, sites.

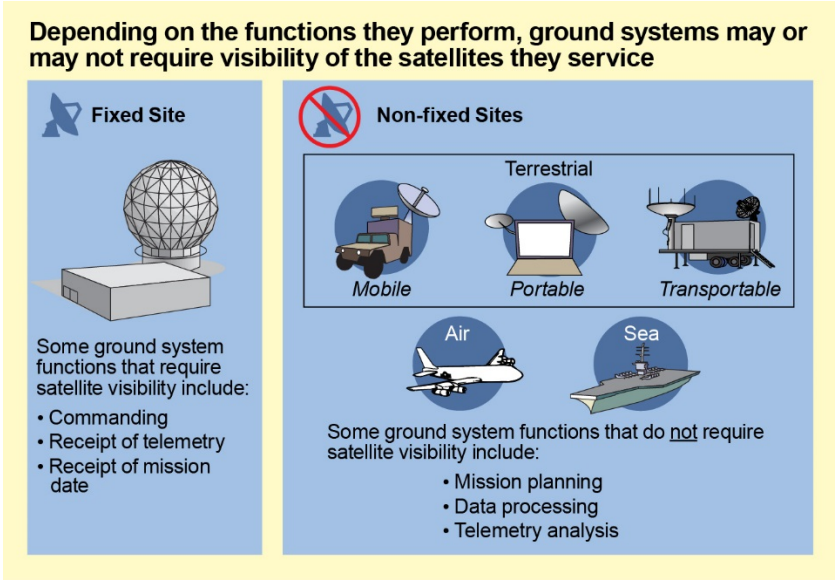


Figure 13-2. Ground facilities taxonomy.

Two classes of considerations drive location(s) selection of satellite ground station(s), categorized as:

- ***Primary:*** considerations and/or requirements required by mission
- ***Secondary:*** considerations that can affect site design and operations but can be mitigated

13.4.1.1 Primary Considerations

Primary considerations include the following parameters, which are solely driven by mission characteristics:

- The orbit regime of the satellite(s) (i.e. are the orbits of these satellites low, medium, or high; nearly circular or highly elliptical, and inclination)
- Number of satellites to be contacted
- Number of required contacts per time period per satellite
- Required view angle to the satellite
- Required contact duration

The satellite's mission requirements drive its orbital characteristics. Four example ground station locations and the visibility footprint of four orbital regimes are shown in Figure 13-3.

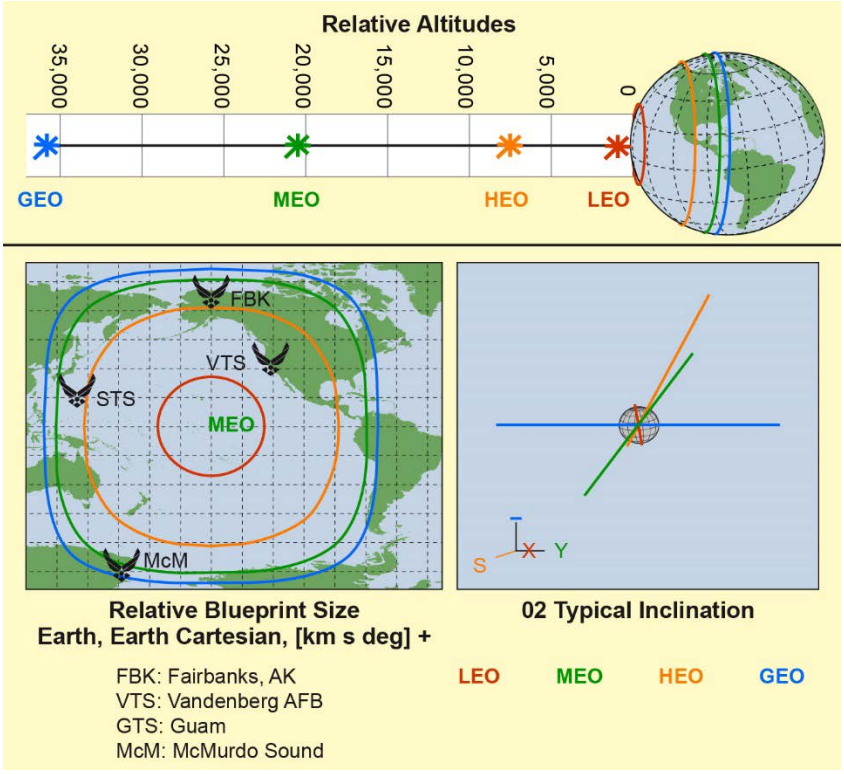


Figure 13-3. Orbital regimes comparison and example ground stations.

Not unexpectedly, satellites that are farther from the surface of the Earth are visible over larger areas. Figure 13-4 shows the contact times between the four example ground stations and satellites with the following orbital geometries:

- LEO60: Low orbit satellite with an inclination of 60°
- LEO98: Low orbit satellite with an inclination of 98°
- HEO: Highly elliptical semi-synchronous satellite
- MEO: Medium altitude semi-synchronous satellite
- SIRIUS: Elliptical synchronous orbit
- GEO45: Elliptical geosynchronous orbit
- GEOS: Geostationary orbit

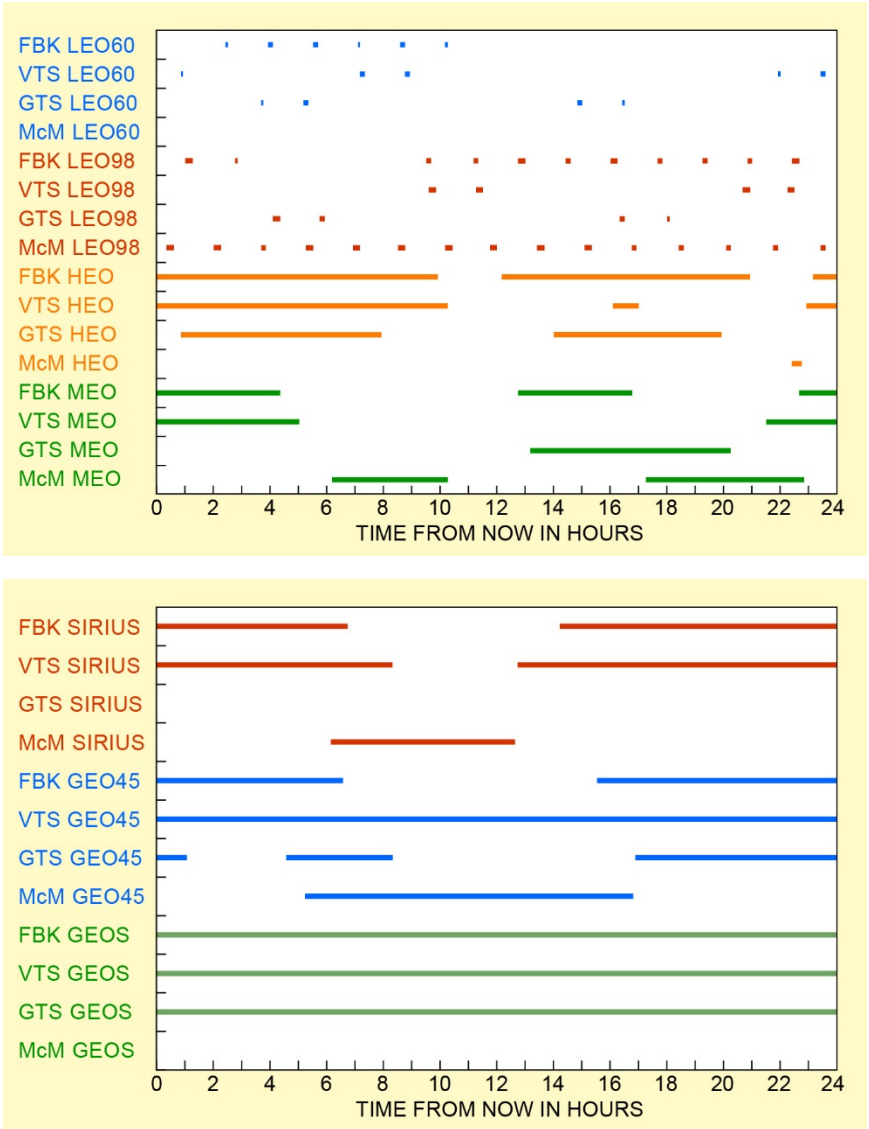


Figure 13-4. Comparison of satellite contact times over a 24-hour period.

The location of the ground station dramatically affects the observation opportunities and durations for the same satellite. The McMurdo station (in the Antarctic) never sees GEOS because the station is so far south that the satellite is always below the horizon. And McMurdo has one brief contact with HEO because that satellite's orbital parameters have been optimized for its Northern

Hemisphere communications mission. The simple way to look at this issue is that locations selected must view all the satellites in the ground station's portfolio frequently enough with enough time per contact for the mission to be accomplished. Software to provide the calculations and graphics to help in this part of the selection process are readily available.

Consequently, orbital regimes impact site design, affecting not only how facilities are placed based on obscura issues, but also dictate requirements related to the capabilities of critical infrastructure, number of ground stations/sites, equipment and terminal (antenna) size, and type. The depiction on Figure 13-5 illustrates the effects of orbital regimes on site design.

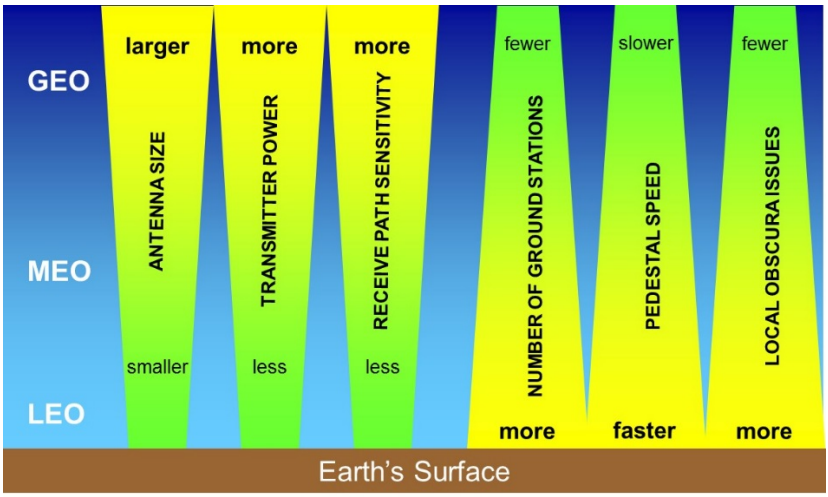


Figure 13-5. Effects of orbital regimes on site design.

13.4.1.2 Secondary Considerations

Secondary considerations can affect site design and operations but the resulting problems posed are not generally "show-stoppers". Secondary considerations can usually be resolved or mitigated by spending additional funds, or by going to an alternate, but similar location. If a suitable ground location is not available, then the use of satellite-to-satellite communications may be an option, but it is one that needs to be considered early in the overall system design planning.

- Political considerations vary, particularly if the site under consideration is located on U.S. soil or in a foreign country. For domestic sites, perceived programmatic advantages in locating a facility in a state or congressional district based on congressional support can affect the funding and/or continuation of the program. While selection criteria

and weighting factors for selecting sites should be focused on impartial evidence, such as technical analyses and cost considerations, if several sites meet all the technical selection criteria but there is some merit in political benefits, then that could play into consideration.

- Although foreign-based sites may be required to meet satellite visibility, they come with burdens beyond those of domestic sites. Aspects of using foreign locations include host country political stability; inspection rights; access to program data; requirements for the use of local personnel or contractors; community attitude; availability and quality of expendables, spare parts, and vendor services; and customs, duties, and taxes. If foreign locations are unavoidable, collocating with an existing U.S. presence is preferred over site development at a new location. In rare cases, it may be possible to use the very limited space at embassies and consulates in foreign countries. Those facilities are legally U.S. soil and are not subject to all of the host country limitations.
- Extreme weather conditions make systems more costly to build, more costly to maintain, and may limit or preclude functionality. Atmospheric moisture causes signal attenuation; the higher the frequency, the more pronounced the attenuation. Cloud cover interferes with optical satellite tracking instruments. High winds disturb pointing accuracy of unprotected antennas and stability of radome structures. Radomes, exposed antennas, and other structures need to account for snow and ice loading. Lastly, systems intended for operation without heat, ventilation, or air conditioning have additional constraints such as operating temperature and humidity.
- High altitude requires special equipment or tailoring of equipment specifications. Specially constructed disk drives are required for elevations over 10,000 feet; air flow required for cooling increases with elevation and the power capacity of generators decreases with altitude.
- Locating a site in an active seismic zone or on permafrost requires special construction techniques that will increase construction costs.
- Legal issues may be impediments to, or may exclude use of, a particular location. Legal obstacles may include encroachment on endangered species habitats and damaging sites of archaeological or historical significance. Working with archaeologically or historically significant sites can often be accomplished but cause unforeseen delays that will extend the schedule and drive up costs.
- Determination of local obscura needs to be understood and evaluated. Objects such as buildings or terrain may block visibility of the satellite at low antenna elevations. A satellite typically requires line-of-sight to the ground station. A non-stationary satellite will subsequently have less contact time per pass due to obscuration, possibly not meeting mission operational requirements.

- Frequency interference issues need to be evaluated. The transmit frequencies and/or receive frequencies may adversely affect nearby users either by interfering with their receivers or overpowering their transmissions. Similarly, nearby emitters may interfere with the site's transmissions and receivers.
- From the site maintenance and operations perspective, the availability of repair parts for equipment and facilities and qualified and accessible vendors for equipment repair, facility repair, janitorial services, and expendables replenishment needs to be evaluated.
- The state of local "infrastructure" has the potential for increasing costs and/or for making acquiring and retaining staff a problem. These factors include the availability, quality, and stability of "commercial" power, communications services, water supply, and sewerage service; roads, airports and their runway length; and local law enforcement and fire suppression capabilities. Available and affordable housing, schools, medical care, and shopping can aid or discourage recruiting of operations and maintenance staff.

13.4.1.3 Site Survey

A critical step in site evaluation and selection is the site survey. Whether the ground station project is a collocation with an existing ground site or is a new site, a visit to the location by a knowledgeable team of subject matter experts is imperative. Important additional information for virtually all of the factors listed in the preceding section can be obtained. In particular, the survey of the local frequency environment should include not only a visit to the local frequency management office if the plan is for collocation, but also taking or obtaining sufficient equipment to listen to and measure the frequency and power of nearby emitters. Visual examination of local obscura should be performed and photographs taken. In addition, investigations should be performed to determine if any future construction nearby could potentially change or block the view of the horizon from the planned antennas locations.

Additional site surveys, including geological and environmental assessments, will be required to thoroughly document the project site area as the project progresses through acquisition.

13.4.2 Threat-Resistive Design and Construction

All facilities face a certain level of risk related to various potential threats. These threats may be a result of natural events (seismic events, tornados, hurricanes, tsunamis, etc.), accidents (explosions, fires, etc.), or intentional acts (terrorism, war, etc.). Regardless of the threat, facilities need to be designed, managed and operated to resist these threats as much as feasibly possible. The security and design of the ground segment facilities should be based on a risk assessment,

performed at the programming phase of a design project, and then periodically re-assessed throughout the lifecycle.

Several federal agencies reference the Federal Protective Security Risk Management (FMSR) process. The DOD and USAF refer to the Unified Facilities Criteria (UFC) for the minimum anti-terrorism standards for design guidance of facilities [4].

13.4.2.1 Threats and Vulnerability Assessments (Risk Analysis)

Threats (credible) evaluation, vulnerabilities assessments, and risk analysis can be applied to any facility or critical infrastructure. Various methodologies and types of assessments exist and have been used for many years. There are also several standards that establish prescriptive and/or performance-based procedures for threats, vulnerabilities and risk assessments.

Once credible threats are identified, a vulnerability assessment should be performed. This evaluation considers the potential impact of loss from a successful attack as well as the vulnerability of the facility and location to an attack. Impact of loss is the degree to which the mission of the agency is impaired by a successful attack from the given threat.

A key component of the vulnerability assessment is properly defining the ratings for impact of loss and vulnerability. These definitions may vary greatly from facility to facility and for the facility service type. For example, the amount of time that mission capability is impaired is an important part of impact of loss. If the facility being assessed is a command and control center, a downtime of a few minutes may be a serious impact of loss, while for an office building a downtime of a few minutes would be minor. A sample set of criteria/definitions for impact of loss is provided in Table 13-1.

Table 13-1. Impact of Loss Ratings and Criteria/Definitions

Rating	Impact of Loss Criteria/Definitions
Devastating	The facility is damaged and/or contaminated beyond habitable use. Most items and assets are lost, destroyed, or damaged beyond repair/restoration. The number of visitors to other facilities in the organization may be reduced by up to 75% for a limited period of time.

Rating	Impact of Loss Criteria/Definitions
Severe	The facility is partially damaged and contaminated. Some items and assets in the facility are damaged beyond repair, but the facility remains mostly intact. The entire facility or portion thereof may be closed for an extended period of time. Assets may need to be relocated for protection. The number of visitors to the facility and others in the organization may be reduced by up to 50% for a limited period of time.
Noticeable	The facility is temporarily closed or unable to operate, but can continue without an interruption for some specified period of time. A limited number of assets may be damaged, but the majority of the facility is not affected. The number of visitors to the facility and others in the organization may be reduced by up to 25% for a limited period of time.
Minor	The facility experiences no significant impact on operations (downtime is less than four hours) and there is no loss of major assets.

Vulnerability is defined to be a combination of the attractiveness of a facility as a target and by the level of deterrence and/or defense provided by the existing countermeasures. Target attractiveness is a measure of the asset (facility) in the eyes of an aggressor and is influenced by the function (and/or symbolic importance) of the facility. Sample definitions for vulnerability ratings are as presented in Table 13-2.

Table 13-2. Vulnerability Ratings and Criteria and Definitions

Rating	Vulnerability Criteria/Definitions
Very High	High profile facility that provides a very attractive target for potential adversaries, and the level of deterrence and/or defense provided by the existing countermeasures is inadequate.
High	High profile regional facility or a moderate profile national facility that provides an attractive target and/or the level of deterrence and/or defense provided by the existing countermeasures is inadequate.

Rating	Vulnerability Criteria/Definitions
Moderate	Moderate profile facility (not well known outside the local area or region) that provides a potential target and/or the level of deterrence and/or defense provided by the existing countermeasures is marginally adequate.
Low	This is not a high profile facility and provides a possible target and/or the level of deterrence and/or defense provided by the existing countermeasures is adequate.

The vulnerability assessment may also include detailed analysis of the potential impact of loss from an explosive or environmental (chemical, biological) attack. Professionals with specific training and experience in these areas are required to perform these detailed and complex analyses. The resulting output of most threat, vulnerability, and risk analyses are recommendations for facilities upgrades and/or safeguards that could be put in place to help mitigate the issues. Safeguards may include procedures, processes, and/or physical means and methods that might be implemented.

Some federal agencies have issued their own security design standards. The most prominent of these are the DOD Unified Facilities Criteria (UFC) UFC 4-010-01*: Minimum Anti-Terrorism Standards for Buildings* [4] and the *Interagency Security Committee (ISC) Security Design Criteria* [5]. Currently, there are no universal codes or standards that apply to both public and private sector buildings. However, it is generally accepted within the architect-engineering industry that security issues must be addressed in concert with other design objectives and integrated into the overall building design throughout the process. This ensures a quality building with effective security. This concept is known as multi-hazard design.

Depending on the building type, acceptable levels of risk, and decisions made (based on recommendations from comprehensive threat, vulnerability, and risk assessments), appropriate countermeasures or mitigations should be implemented to protect people, assets, and mission. Types of attack and threats to consider include (not all-inclusive):

- Unauthorized entry (forced and covert)
- Insider threats
- Explosive threats: Stationary and moving vehicle-delivered, mail bombs, package bombs
- Ballistic threats: Small arms, high-powered rifles, drive-by shootings, etc.
- Weapons of mass destruction (chemical, biological, and radiological)
- Cyber and information security threats

Critical infrastructure protection surveys (CIP Surveys) are used to assess vulnerabilities and potential single points of failure of ground stations and their critical infrastructure.

13.4.2.1.1 Unauthorized Entry (Force Control)

Protecting the facility and assets from unauthorized persons is an important part of any security system. Some items to consider include:

- Compound or facility access control
  - Control perimeter: Fences, bollards, anti-ram barriers
  - Traffic control, remote controlled gates, anti-ram hydraulic drop arms, and hydraulic barriers, parking
  - Forced-entry-ballistic resistant doors and windows
- Perimeter intrusion detection systems
  - Clear zone
  - Video and CCTV
  - Alarms
  - Detection devices (motion, acoustic, infrared)
- Personnel identification systems
  - Access control, fingerprints, biometrics, ID cards
- Protection of information and data
  - Acoustic shielding
  - Shielding of electronic security devices from hostile electronic environment
  - Secure access to equipment, networks, and hardware, e.g. satellites and telephone systems

13.4.2.1.2 Explosive Threats (Blast/Bomb)

Federal standards and criteria are widely recognized as the primary source of guidelines for the design of buildings to resist explosive threats. Because of the uniqueness of each building's mission, functional requirements, and physical security design objectives, there are limited codes and standards that apply to blast mitigation design.

Explosive threats tend to be the criminal and terrorist weapon of choice. Devices may include large amounts of explosives that require delivery by a vehicle. However, smaller amounts may be introduced into a facility through mail, packages, or simply hand carried in an unsecured area. Normally the best defense is to provide defended distance between the threat location and the asset to be protected. This is typically called standoff distance [4]. If the distance is not available or is insufficient to reduce the blast forces reaching the protected asset, structural hardening may be required. If introduced early in the design

process, this may be done in an efficient and cost-effective manner. If introduced late in a design, or if retrofitting an existing facility, adding this type of measure may prove to be economically unfeasible and therefore create a risk. Some items to consider include:

- Ensuring the design team includes qualified security and blast consulting professionals.
- Providing defended standoff with rated or certified devices that meet antiterrorism force protection requirements.
- Considering structural hardening and hazard mitigation designs.
- Designing the facility with redundant egress and other critical infrastructure to facilitate emergency evacuation and control during an event.

13.4.2.1.3 Cyber and Information Security Threats

Business continuity of operations and mission function rely heavily on the transmission, storage, and access to a wide range of electronic data and communication systems. With the evolution of wireless technology and SMART (intelligent automation systems) building applications, facilities may become more vulnerable to cyber-attacks. Protecting these systems from attack is critical. Some items to consider include:

- Understanding and identifying the information assets that need protection.
- Protect the physical infrastructure that supports information systems.
- Implement monitoring devices to detect and prevent unauthorized access to or destruction of sensitive information.

13.4.3 Data Processing Operations Considerations (and Challenges)

Ground station functions are supported by communications equipment that are stacked in racks located typically in rows of some organization within data center(s). These data centers are critical to the monitoring, command, and control of satellites and spacecraft as well as the processing of satellite data. The data center can be small, occupying a single room, or expansive, occupying an entire building. Space, power, and cooling are the critical triad supporting the communications backbone of the ground station data center.

The Telecommunications Industry Association (TIA), a trade association accredited by the American National Standards Institute (ANSI), publishes ANSI/TIA-942: *Telecommunications Infrastructure Standard for Data Centers* [6]. This standard defines the quantifiable tiers used to design data centers and

requirements for the data center infrastructure. The Uptime Institute, a consulting firm, has also published its own independent guidelines describing four tiers based on availability; the following summarizes the four tiers and the respective data center infrastructure characteristics: [5]

- Tier I—Basic Data Center: Composed of a single path for power and cooling distribution, without redundant components (99.671% availability)
- Tier II—Redundant Components: Composed of single path for power and cooling distribution, with redundant components (99.741% availability)
- Tier III—Concurrent Maintainable: Composed of multiple active power and cooling distribution paths (only one active at a given time) with redundant components, allowing for concurrent maintainability (99.982% availability)
- Tier IV—Fault Tolerant: Composed of multiple active power and cooling distribution paths, with redundant components, and is fault tolerant (99.995%)

It is important to note that merely having redundant components do not preclude disruptions to the system; the overall system configuration needs to be designed/validated to ensure expectations are met.

The Aerospace Corporation published a technical operating report (TOR) titled, "*High Density Data Design Guidelines*" which outlines a concept for organizaing and designing a typical data center [8].

Architectural space and layout: Overall system design and data center layout has a much greater impact on the effeciveness of the data center than the efficiency of the individual components. The data center floor layout contributes more to the efficiency of the design than the selection of individual power and cooling devices. A primary concern in the layout of a data center floor layout is the segregation of cold and hot air coupled with exhaust and airflow. Fundamentally, better data center performance comes down to thermal (heat) management.

Business continuity: Continuity of operations is a main concern and needs to be considered in the design (and/or upgrades/modifications) of the data center. It is essential to provide a reliable critical infrastructure for its operations, typically accomplished through redundancy of mechanical cooling (HVAC) and power systems including generators and uninterruptible power sources (UPS) and controls.

Electrical power: Considerations should emphasize maximum system scalability, flexibility, and redundancy. As a minimum, the system should provide for [8]:

- Allowances for performing concurrent maintenance on any portion of the electrical supply without interruption to the end user.
- Continuity of electrical service after a single failure of any piece of electrical equipment.

Mechanical systems (environmental): Considerations should emphasize maximum system scalability, flexibility, and efficiency. As a measure of efficiency, air-side economizers should be considered and integrated into the design when applicable [9]. The key components of this system are the ability to expand the system only as required by rack deployment schedules. As a minimum, the mechanical systems should provide for [8]:

- Allowances for performing concurrent maintenance on any portion of the system without interruption to any end-user rack.
- Continuity of mechanical services (cooling) after a single failure of any piece of mechanical equipment.

Security: The security of a data center is crucial and must take into account physical security, network security, and data and user security.

Scalability: It is important when planning a data center, to anticipate and plan for potential future requirements. Modularity and flexibility are key elements in enabling these data centers to expand and adapt as new requirements or functions are added, as demanded by the mission.

13.5 Programmatic Considerations

Early investment in planning, programming, and concept design of ground facilities and infrastructure can help avoid unnecessary costs and delays. An effective process balances the key project constraints by providing a decision-making tool throughout the project based on the customer's needs/values, performance metrics, established procedures, and project goals (Figure 13-6). It implements management systems and processes that control changes while maximizing investment. Participation of all vested parties and stakeholders in front-end strategic planning and requirements definition activities is key.

Figure 13-6. Balanced programming approach.

13.5.1 Requirements Definition (Architectural Programming)

Architectural programming defines the process by which the customer's or program's purpose, goals, and needs are translated into a desired result or project. The project is successful when the design, construction, and operations of the facility/building meet the functions it was intended to provide. The qualities of a successful project may not even be noticed or recognized, but a poorly functioning building can be costly to correct, if even possible. There are three overarching principles associated with ensuring functional building design and operations:

- Establish functional needs
- Validate design and systems integration
- Verify attainment of performance objectives

Spatial needs are a primary element of the planning process that translates an owner's spatial, mission needs, and service performance requirements to a facility program. This process seeks to: state the problem, establish goals, collect and analyze facts, establish functional relationships, uncover and test concepts, and determine needs.

Adopting a methodical approach that extends through all phases of a project, from pre-design through owner occupancy and operation (to disposal), with checks (gates) at each stage of the process helps ensure validation that decisions and designs meet the owner's program and design requirements.

Poorly defined programmatic requirements are by far the biggest issue with verifying that design and performance requirements have been met. To help determine and validate requirements, a successful project addresses the following actions (at a minimum):

- Understanding how the work processes support the mission and purpose of a facility; also known as the concept of operations (CONOPS)
- Defining spatial requirements for occupant activities and equipment
- Understanding functional relationships between program spaces
- Defining the operational parameters and performance criteria for the facility and critical infrastructure

13.5.1.1 Integrated Project Approach

Successfully designed facilities, which function properly in all aspects, typically implement an integrated project approach. The process typically implements an integrated team comprised of all stakeholders across the lifecycle of the project process and into operations, using an interactive approach to the design/construction process. Everyone involved in the planning, design, use, construction, operation, and maintenance of the facility work together to understand the issues and concerns and interact closely throughout all phases of the project.

One technique used to gain stakeholder buy-in is using a focused and collaborative brainstorming session held at the beginning of a project. This encourages an exchange of ideas and information, allowing integrated design solutions to take form. This technique, also known as a "charrette", is particularly helpful in complex situations where many people represent the interests of a client/program and there are potential conflicting needs or multiple groups and agencies perspectives. Participants are educated about the issues, and resolution enables them to "buy into" the schematic solutions. Final solutions or concepts are not necessarily produced, but important and often interdependent issues are explored.

The integrated project approach works to integrate the selected building systems, materials and components, and technologies along with the programmatic mission needs to be mutually supportive as a cohesive and integrated system.

13.5.1.2 Building Codes, Standards and Regulations, and Federally Mandated Requirements

The following list outlines the compliance and reference documents that are most applicable, but it is not meant to be all-inclusive:

UFC	Unified Facility Criteria	UFC documents provide planning, design, construction, sustainment, restoration, and modernization criteria, and apply to the military departments, the defense agencies, and the DOD field activities in accordance with DOD Directive 4270.5 (Military Construction)
UFC 1-200-01	Unified Building Criteria: General Building Requirements	Provides general building requirements, establishes use of consensus building codes and standards, identifies key core UFC, and identifies unique military requirements. (Military Construction)
IBC	International Building Code	The model code developed by the International Code Council (ICC) and adopted throughout the majority of the United States as the applicable building code. State codes may also apply, depending on the jurisdiction. There are additional sub-codes that apply (I.e., International Plumbing Code, International Fire Code, etc.)
40 CFR	Environmental Protection Agency	Environmental guidelines and regulations
OSHA	Occupational Safety and Health Standards	Safety guidelines and regulations
ADA/USC 12181	Americans with Disabilities Act	Provisions and guidance
ASHRAE Handbook	American Society of Heating, Refrigerating, and Air-Conditioning Engineers	Series of standards and guidelines related to HVAC systems and issues; Aligns with the UFC and IBC
NFPA	National Fire Protection Association	Fire protection codes; Aligns with the UFC and IBC
NEC	National Electrical Code	Electrical codes; Aligns with the UFC and IBC
MIL-HDBK-419/1A	Grounding, Bonding and Shielding for Electronic Equipment and Facilities, Volumes 1 and 2; Theory and Application	Considerations for design, construction, operation, and maintenance of electronic equipment (and facilities)

MIL-STD-1472F	Human Engineering Design Criteria for Military Systems, Equipment and Facilities	Establishes general human engineering criteria for design and development of military systems, equipment (and facilities)
MIL-STD-1542B	Electromagnetic Compatibility and Ground Requirements for Space System Facilities	Specifies design, performance, and verification requirements for electrical subsystems for space systems, including EMC, electrical power grounding, bonding (and TEMPEST security)
Facilities Standards		Facilities standards establishes standards and criteria for new buildings and alterations to existing buildings. Each military base typically has an established and adopted standard specific to the installation. Additionally, the intelligence community (IC) may also have an adopted standard.
IC Tech Spec for ICD/ICS 705		Technical specifications for construction and management of sensitive compartmented information facilities
ICD 503		Intelligence community information technology systems security risk management, certification, and accreditation
ETL 01-18	Fire Protection Engineering Criteria	Electronic equipment installations
ETL-01-1	Reliability and Maintainability (R&M)	Design/maintenance Checklist
LEED Standard	Leadership in Energy and Environmental Design Standards	Provides benchmark (rating systems) for high-performance green building design, construction, operation, and maintenance.
USGBC	U.S. Green Building Council	Provides information and guidelines to LEED project registration and certification requirements. (www.usgbc.org)

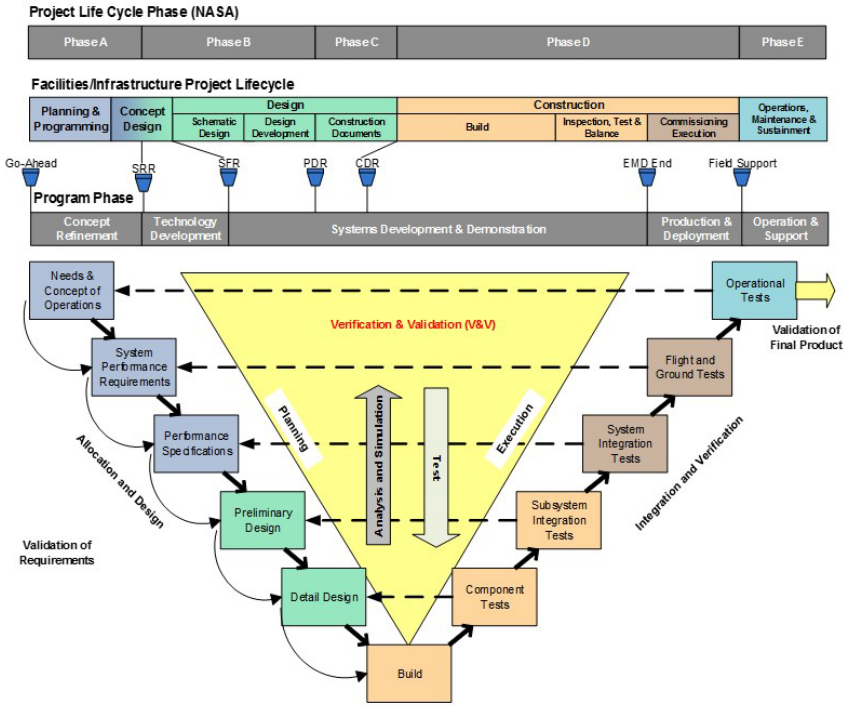
13.5.2 Acquisition Process

13.5.2.1 System Lifecycle

The acquisition of a major space system is accomplished through phases that constitute the system lifecycle. These phases are sequentially designated:

- Pre-concept
- Concept exploration (schematics)
- Demonstration and validation
- Engineering and manufacturing development
- Production and deployment

The phases provide the common timeline (language) for implementing activities related to acquisition and its support components. In the facilities design, construction, and operations realm, these phases are identified by a different nomenclature (taxonomy) and slightly different timeline but accomplish a similar purpose. The Figure illustrates the timeline of the various phases in the lifecycle of a facility as aligned with the validation and verification processes for a major space program lifecycle (program phase SRR, PDR, CDR, etc.), the project life phase typically tracked by NASA (Phase A, Phase, B, etc.), and the systems engineering validation and verification (V&V) process flow [11].



Source: Adapted from USC SAE 541: Systems Engineering Theory and Practice: Verification Validation, Test & Transition; Jim Hines, Spring 2010

Figure 13-7. Comparative V&V process flow alignment.

In Figure 13-7 there are essentially four major phases with a fifth phase, concept design, straddling between planning and design activities within the facilities and infrastructure project flow. With each phase, there are a series of activities and consequential outputs that allows the project to advance to the next step or gate.

Within the context of each phase, there are a series of activities related to development, engineering, validation, and verification that are supported by controls, enablers, and inputs. These process flows are outlined in Figure 13-8 through Figure 13-13.

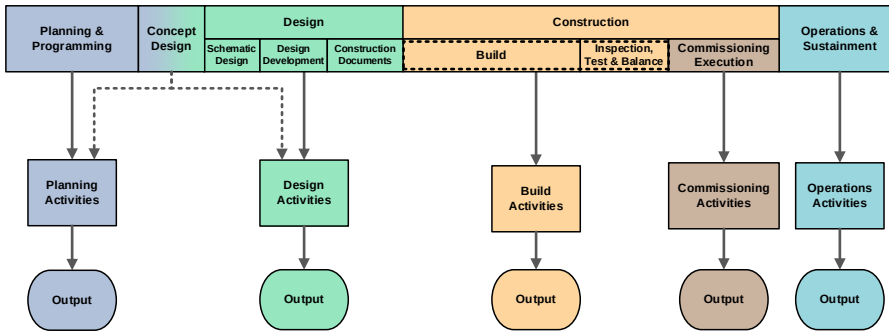


Figure 13-8. Facilities/infrastructure project flow.

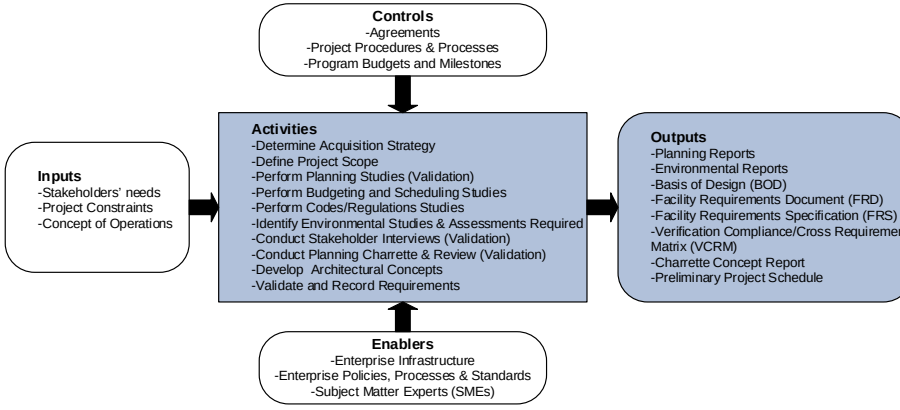


Figure 13-9. Context for the planning and programming process.

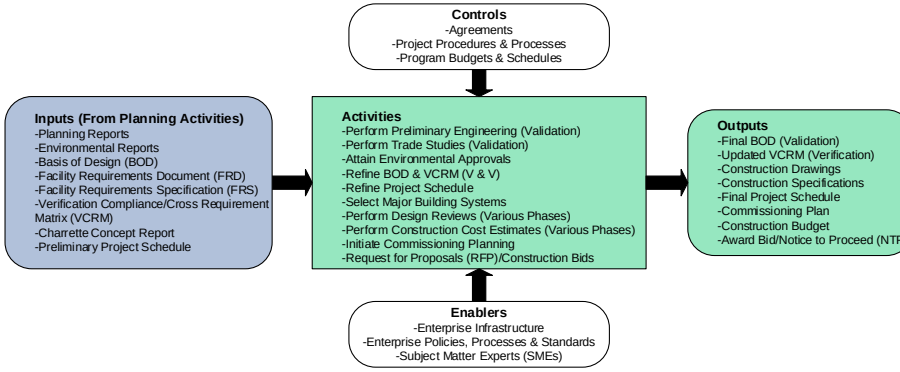


Figure 13-10. Context for the design process.

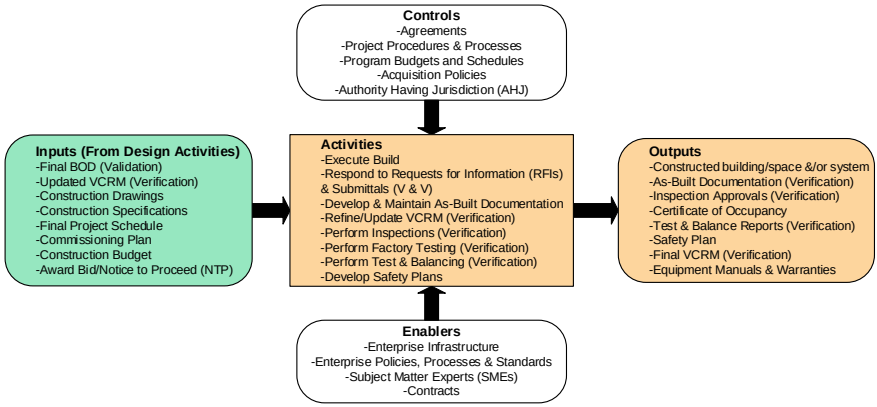


Figure 13-11. Context for the construction process.

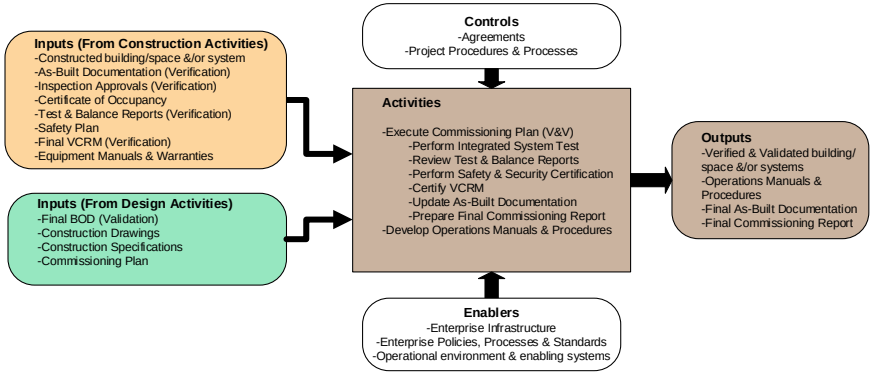


Figure 13-12. Context for the commissioning process.

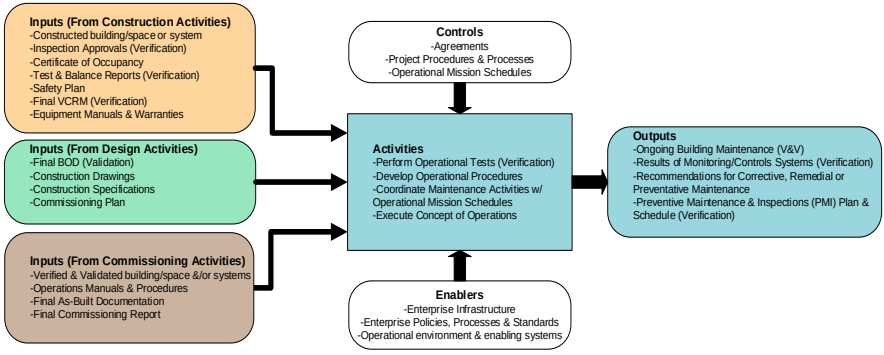


Figure 13-13. Context for the operations and sustainment process.

Development of the ground system is the beginning and end of all space operations. Involvement of stakeholders and facility users/managers in all phases of the system life cycle is essential for program success. The ground system infrastructure is a relatively minor cost element upfront compared to the total expenditure for a space system. Failure to address it in the early phases, however could add significant cost and schedule consequences to the project/facilities over the long-term, especially given the fact that facilities many times outlive the program they were built to serve. The most cost-effective time to make changes is during the programming phase, as demonstrated in Figure 13-14. This phase is the best time for stakeholders (interested parties, etc.) to influence the outcome and results of the project. Implementing an integrated approach to meeting mission CONOPs and related facilities/infrastructure is essential. The primary point is that the facilities and related infrastructure need to be considered as part of the space system, not an adjunct, and they need to be considered early in the space systems planning process.

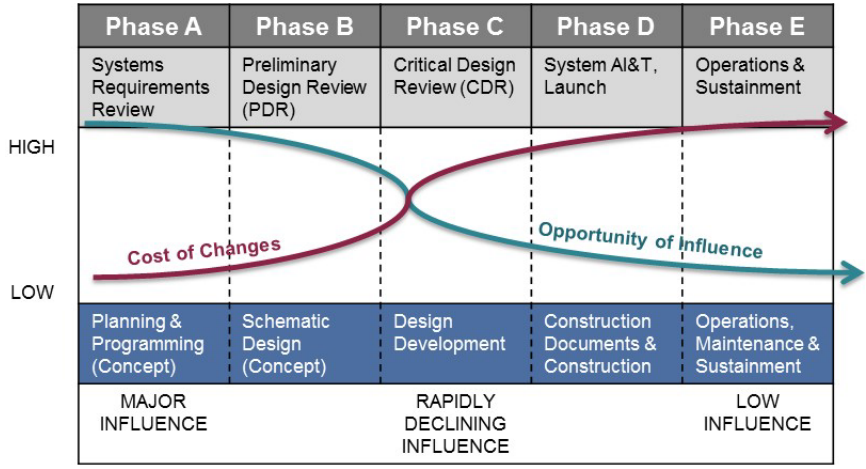


Figure 13-14. Cost versus opportunities in making changes.

13.5.2.2 Project Delivery Methods and Controls

A project delivery method defines the relationships, roles, and responsibilities of parties and the sequence of activities required to design and construct a facility. Several delivery methodologies exist, but the three major delivery systems are:

- ***Design-Bid-Build:*** This is the traditional approach where the owner contracts with a design agent to provide complete design documents and then contracts separately with a contractor to construct the facility in implementing a sequential phase implementation.

- ***Construction Management at Risk:*** This approach is similar to the traditional design-bid-build method but the design and construction phases are overlapped to allow for value engineering and constructability input during the latter part of the design phase from a construction manager.
- ***Design-Build:*** This is an approach in which the owner contracts with a single entity to perform both the design and construction under a single design-build contract with a single point of responsibility for both.

There are benefits and drawbacks to each of the above methodologies depending upon the project application. Several factors should be considered before a project delivery system is adopted. In recent years, particularly since acquisition reform, the federal government has leaned towards design-build as the favored approach. The reason behind this initiative is the perception that money will be saved. However, if design-build is applied incorrectly, the methodology can potentially affect costs adversely. In general, design-build as an approach is not well-suited for:

- Highly complex projects;
- Unique, one-of-a kind projects with special design requirements;
- Facilities with clients that need lots of attention (and those with lots of regulatory impositions); or
- Projects requiring flexibility implementing innovative construction methodologies.

The above items are just some of the considerations in determining the approach in designing and constructing ground station facilities. There are several aspects that go into building acquisition decisions. Some predominant factors that affect project success:

- Owner type (public vs. private)
  - Owner's representative's capability (and authority)
  - Owner's ability to define project scope
  - Owner's ability to make timely decisions
- Project complexity
- Level of design complete prior to engaging construction team
- Regulatory/legal constraints on the project
- Project team selection
  - Ability to restrain contractor pool (pre-qualification)
  - Labor type (union vs. non-union split)
  - Contractor's work split (direct hire vs. subcontracted)
- Individual experience of team members
  - With the project delivery system chosen
  - With similar facility type

    - Prior experience working as a team unit
- Project team communication

There are some key, top-level controls that support execution of a successful design/construction project, regardless of the type of project delivery method used. These controls are as follows:

- Scope management
- Cost management
- Schedule management
- Quality control (configuration management)
- Building commissioning (testing and quality assurance)

The first four are common to most projects of any type. Building commissioning, the fifth item, is specific to facilities and therefore included in discussion as part of this chapter. Similar to testing and quality assurance processes in the aerospace industry, building commissioning provides the framework for ensuring that facility performance objectives are met. (See 13.5.2.2.1)

13.5.2.2.1 Building Commissioning (Verification and Quality Assurance)

Meeting performance objectives is achieved by a sustained effort, from inception and planning through turnover and operation, to assure the delivery of a project that satisfies all of the owner's functional and operational requirements. Building commissioning is essentially the verification and quality assurance process for facilities. Building commissioning is an all-inclusive, quality assurance-based process for working with the project teams and documenting the planning, delivery, verification, and risk management as the project moves from one phase and transitions to another. The process is interdependent with requirements development, providing better building documentation (verification of compliance and acceptance).

The overarching goal of commissioning a project is to define that there is a clear mission for the building and to obtain confirmation that the building works as intended to fulfill that mission. Expected commissioning outcomes include:

- Delivering a facility that meets owner's needs and requirements;
- Verifying of compliance with codes, standards, and all regulations
- Preventing and/or mitigating potential issues through proactive processes;

- Providing better documentation and records related to: design, construction, testing, and maintenance requirements for sustaining performance through the lifecycle of the facility;
- Obtaining baseline performance metrics that enable future trending (performance, etc.); and
- Potentially lowering costs through the lifecycle.

Implementing a building commissioning plan and process helps confirm that all energy-related systems are functioning at their required efficiency levels. This verifies that performance requirements have been met, optimizes energy use, reduces operating costs, and confirms environmental conditions and performance.

13.5.2.3 Military Construction (MILCON) Fundamentals

The Federal Acquisition Regulation (FAR) has prescribed funding requirements and limitations related to design/construction of military construction projects and also facilities, sustainment, restoration, and modernization (FSRM) processes. There are many details that need to be considered by program managers and facilities planners (architects-engineers) during these processes, including submissions of concept proposal packages (Form DD 1391). Examples of the MILCON and FSRM funding limits are provided in Table 13-3 (refer to reference cited for most current limits).

Table 13-3. MILCON and FSRM Funding Limitations (Reference Only)

Program	Category	Fund Range	Approval Authority
Locally Approved Projects	Maintenance Repairs (M1) Minor construction (R1)	$300K–less $100K–less	Installation commander
Special Projects	Repair (M2) Construction (R2) Construction (R2) Repair (R2) Repair (R2)	$300K–$5M $100K–750K $750K–$1.5M $5M–$7.5M $7.5M–above	HQMC HQMC ASN/Congress ASN ASN/Congress
MILCON Projects	Construction	$750K–above	Congress

Program	Category	Fund Range	Approval Authority
MILCON Exceptions	Unspecified minor construction (UMC)	$750K–$1.5M or $1.5M–$3M	HQMC ASN/Congress
	Emergency construction	$750K–Less	ASN/Congress
	Restoration of damaged facilities	$750K–Less	ASN/Congress
	Contingency construction	$750K–less	ASN/OSD/Congress
Interim/Re-locatable	Trailers, modulars, tension fabric structures, etc.	$250K–below	HQMC
		$250K–above	HQMC

NOTES:

Limit is $1.5M for construction projects solely to correct a deficiency that is life, health, or safety-threatening.

UMC limit is $3M for construction projects solely to correct a deficiency that is life, health, or safety-threatening.

The MILCON approval process requires methodical planning and time. The MILCON process flow and timelines are illustrated in Figure 13-15 at a very top level. Depending on the type of project delivery approach, as discussed in 13.5.2.2, and scope, the project timeline from initiation of planning to project completion requires between three to five years, on average. For large or complex projects, it is not uncommon for the timeline to be between five and ten years.

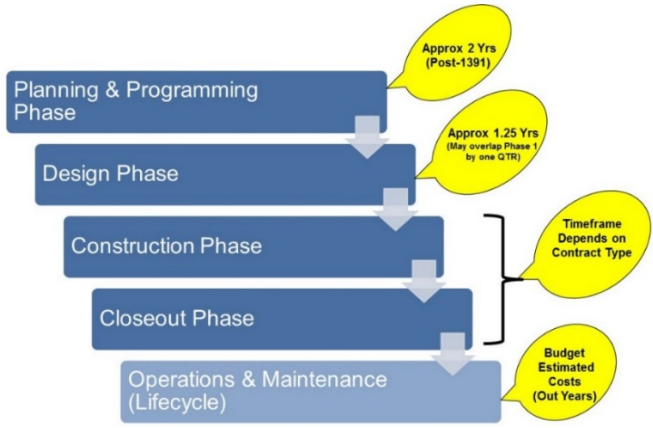


Figure 13-15. MILCON flow and estimated process time.

There are several related activities and submissions (document deliveries) that occur within the design and construction phases of the project and at phase transitions within the MILCON flow (Figure 13-16). For MILCON projects. As required with MILCON projects, congressional notification occurs at the end of the planning and programming phase and prior to initiation of the design phase. All this needs to be accounted for in the project schedules.

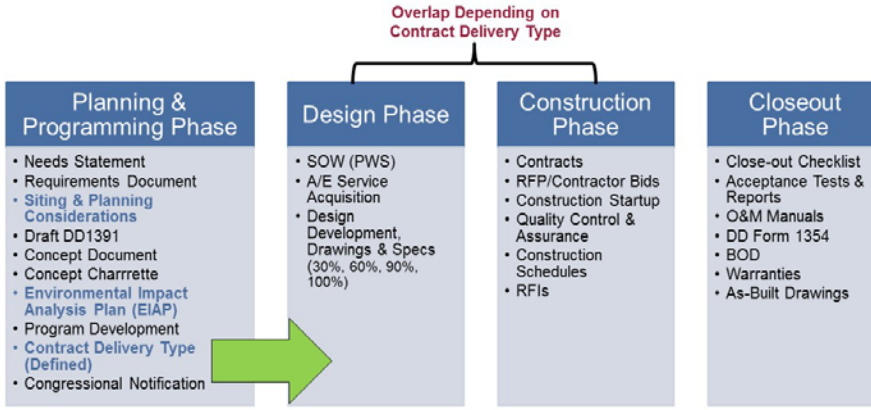


Figure 13-16. Project submissions (per phase).

13.5.3 Operations, Maintenance, and Sustainment (OM&S)

Although the front-end cost of facility design and construction may not be significant when compared to the total cost of a satellite program, the costs associated with the operations, maintenance, and sustainment of the facility over its lifecycle have the potential to become a significant cost burden. Decisions made on the project front-end during acquisition through design and construction have significant potential towards affecting costs on the back-end, once the facility is into operations. Given the propensity for buildings to be asked to perform beyond their design-life and even beyond the program's life, this has the potential for added impact. When making acquisition and design trade-offs and decisions, it is very important that potential impacts to OM&S be considered and factored in the true cost of a project. Commissioning, as discussed in 13.5.2.2.1, is also a significant consideration for OM&S lifecycle cost savings. Related planning should be included in the front-end project planning activities and continue through the project phases to operations. It is critical to consider the CONOPs of how the facility and systems will be used and then design and test for performance accordingly. On existing facilities, retro-commissioning might be considered to determine needed upgrades or modifications to the system to improve efficiencies, reliability, and availability.

An example is the operational need driving constant availability demand with no down time. Unless the facility systems are designed and configured to allow for concurrent maintenance, the ability to perform maintenance typically becomes extremely limited or non-existent. Upgrades or replacement of critical components due to end of life are delayed, sometime indefinitely. Consequently, failures end up driving maintenance and/or upgrades.

13.6 Emerging Technologies and Future Considerations

The workplace is changing in response to rapid technological advances and, consequently, facilities will need to be able to respond to these changes. Facilities and related infrastructure will need to be designed and constructed to be strategic, flexible, and agile, especially given that the typical lifecycle of facilities spans over several decades. Given that the federal acquisition system tends towards short-sightedness (planning/budgeting year-to-year), there are respective challenges to overcome.

13.7 References

1. DOD, "Dictionary of Military and Associated Terms," Department of Defense, Washington D.C, 2001 (2005 Revision).

2. Benator, Sheri, Mel Cutler, Marilyn Dubas, Jim Shneer, David Bart. *Ground System Architecture Workshop for Satellite Operations Primer and Acquisition Considerations,* The Aerospace Corporation, El Segundo, CA. 2006.

3. Interagency Security Committee. *The Risk Management Process for Federal Facilities: An Interagency Security Committee Standard*. August 2013. U.S. Department of Homeland Security, Washington, DC.

4. Department of Defense, *Unified Facilities Criteria (UFC) Program.* www.wbdg.org/references/pa_dod.php.

5. (FOUO) Department of Homeland Security. *Interagency Security Committee Security Design Criteria*. Washington, DC.

6. Telecommunication Industry Association. *ANSI-TIA-942 Telecommunication Infrastructure Standard for Data Centers*. April 2013.

7. Turner, W. Pitt, Joh H. Seader, Vince Renaud, Kenneth G. Brill. *The Classifications Define Site Infrastructure Performance*. Santa Fe, NM. 2008.

8. T. A. Corporation, *High Density Data Design Guidelines*, TOR-2008(3902)-7782. The Aerospace Corporation, El Segundo, CA. 2008.

9. S. Escalante, *Use of Air-side Economizers for Data Center Cooling*, ATM-2010(3914-10)-1. The Aerospace Corporation, El Segundo, CA. 2010.

10. J. Hines, *Systems Engineering Theory and Practice: Verification, Validation, Test & Transition*. University of Southern California SAE 541. Los Angeles, CA. 2010.

11. Sanvido, Victor and Mark D. Konchar. *Selecting Project Delivery Systems: Comparing Design-build, Design-Bid-Build, and Construction Management at Risk*. The Project Delivery Institute, State College, PA. 1999.

12. J. D. Castro-Bran and S. M. Escalante, *Operations and Facilities Engineering: Requirements Development, Validation & Verification*," The Aerospace Corporation, El Segundo, CA. 2013.

13. *Unified Facilities Criteria (UFC): DOD Minimum Anti-terrorism Standards for Buildings*. Department of Defense, Washington, DC. October 2013.

14. *ISC Security Design Criteria for New Federal Office Buildings and Major Modernization Projects*. National Academies Press, Washington, DC. 2003.

15. ANSI/TIA-942-2005. *Telecommunications Infrastructure Standard for Data Centers*. Telecommunications Industry Association, Arlington, VA. April 2005.

16. Air Force Instruction 32-1032. *Planning and Programming Appropriated Fund Maintenance, Repair, and Construction Projects*. October 2014.

13.8 Acronyms

AHS	authority having jurisdiction
AI&T	assembly, integration, and test
ANSI	American National Standards Institute
BOD	basis of design
CCTV	closed circuit television
CDR	critical design review
CIP	critical infrastructure protection
CONOPS	concepts of operation

DOD	Department of Defense
EIAP	environmental impact analysis plan
FAR	federal acquisition regulation
FE-BR	forced entry-ballistic resistant
FMSR	federal protective security risk management
FRD	functional requirements document
FSRM	facilities, sustainment, restoration, and modernization
GEOS	geostationary orbit
GS	ground segment
HEO	high earth orbit
HVAC	heating, ventilation, and cooling
ID	identification
LEO	low Earth orbit
LS	launch segment
MCC	mission control center
MCN	maintenance control number
MEO	medium Earth orbit
MILCON	military construction
NASA	National Aeronautics and Space Administration
NTP	notice to proceed
O&M	operations and maintenance
OM&S	operations, maintenance, and sustainment
PDR	preliminary design review
PMI	preventative maintenance and inspections
RFP	request for proposal
SMART	intelligent automation systems
SME	subject matter experts
SOW	statement of work
SRR	system readiness review
SS	space segment
TIA	telecommunications industry association
UFC	unified facilities criteria
UMC	unspecified minor construction
UPS	uninterruptible powersource
US	user segment
USAF	U. S. Air Force
V&V	verification and validation
VCRM	verification cross-reference matrix

Chapter 14
Ground Segment Development Cycle

Suellen Eslinger
Software Engineering Subdivision
Computers and Software Division

14.1 Introduction

This chapter summarizes the overall acquisition lifecycle, ground segment acquisition lifecycles, ground segment development lifecycles, and ground segment development phases. The ground segment development phases consist of concept development, requirements, architecture and detailed design, product development, test planning and execution, transition to operations, and maintenance and sustainment. This chapter also discusses stakeholder roles and responsibilities, identifies key acquisition and development products, and provides a list of applicable resources.

Note that the figures in this chapter depicting various lifecycles are not to scale in time.

14.2 Definitions

Acquirer An organization that procures products for itself or another organization. The acquirer team includes the acquirer personnel as well as Federally Funded Research and Development Center (FFRDC), Systems Engineering and Technical Assistance (SETA), and Systems Engineering and Integration (SE&I) contractors [1].

Build A version of the system or software that meets a specified subset of the requirements that the completed system or software will meet [1]. Contractors use various synonyms for the term "build," such as "increment," "release," "block," "iteration," "cycle," "drop," and "spiral." In this chapter, the term "build" will be used for software development iterations, and the term "increment" will be used for the ground segment development iterations that may include hardware, software, and facilities.

CDRL item The term is used for the list of deliverable data in a contract. Data here refers to information, usually described in documents. An individual deliverable data product is called a CDRL item. The CDRL does not include the hardware or software to be delivered for the ground segment and installed in the operational facility.

Developer An organization that develops products ("develops" includes new development, modification, integration, reuse, reengineering, maintenance, or any other activity that results in products) for the contract. The term "developer" encompasses all contractor team members [1].

Hardware item An aggregation of hardware that satisfies an end-use function and is designated for specification, interfacing, qualification testing, configuration management, or other purposes [1].

Increment See "build."

Lifecycle model A project management framework providing a sequencing strategy and a disciplined approach to the structure and order of activities

Software item An aggregation of software that satisfies an end-use function and is designated for specification, interfacing, qualification testing, configuration management, and other purposes. Software items are selected based on tradeoffs among software function, size, host or target computer systems, developer, support strategies, plans for reuse, criticality, interface considerations, the need to be separately documented and controlled, and other factors. A software item is composed of one or more software units. A software item is sometimes called a computer software configuration item (CSCI) [1].

Stakeholder A group or individual that is affected by or is in some way accountable for the outcome of a product [2].

Waterfall A system or software lifecycle model in which the constituent activities are performed in sequential order, possibly with overlap but with little or no iteration. For development, the activities typically include a concept development phase, requirements phase, architecture and design phase, product development phase, test planning and execution phase, transition to operations phase, and maintenance and sustainment phase [1].

14.3 The Acquisition Lifecycle

The system program office's acquisition team and the contractor's team must function within the overall acquisition lifecycle dictated by the acquisition organization's policies and regulations. Within this structure, the contractor team uses a development lifecycle dictated by their internal policies and processes. The acquisition and development lifecycles must fit together in an efficient and effective manner to obtain an operationally viable and sustainable system that meets its requirements within its cost and schedule constraints. Figure 14-1 depicts the relationships of the acquirer and developer activities before and after contract award.

Most acquisition organizations have one or more acquisition lifecycles defined in an organization's policies. This chapter uses the Department of Defense (DOD) acquisition lifecycles defined in the DOD instruction DODI 5000.02, *Operation of the Defense Acquisition System* [3] to illustrate the concepts under discussion. Because each acquisition organization has its own set of approved acquisition lifecycles, and because acquisition policy for all organizations is subject to frequent change, the phases from the *Mission Assurance Guide (MAG)* [4] can be used to help understand how other acquisition lifecycles relate to the DOD acquisition lifecycles shown as illustrations in this chapter. The MAG phases, along with alternative phase names for ground segment acquisition where applicable, are listed in Table 14-1.

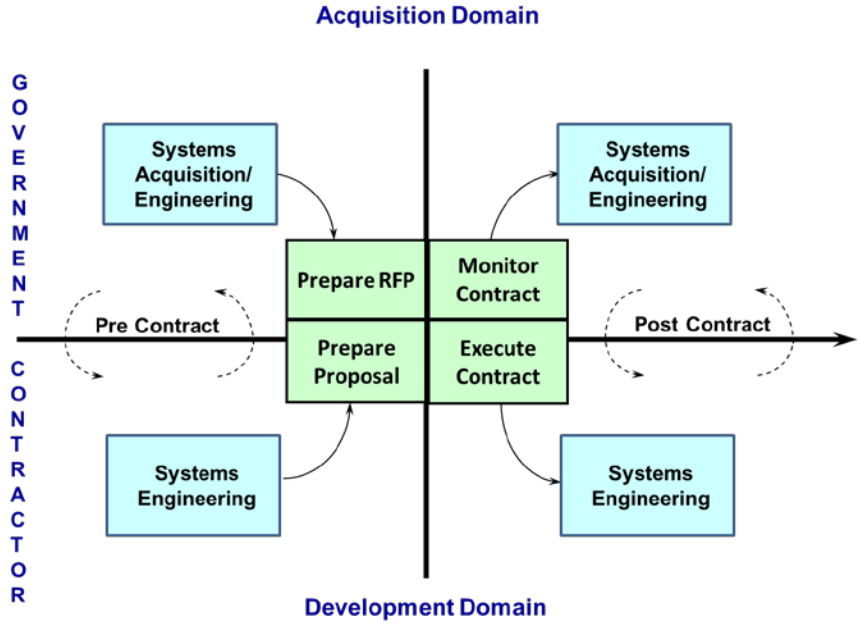


Figure 14-1. Acquisition and development domains.

Table 14-1. Mission Assurance Guide Phases

MAG Phases	MAG Phase Names	Alternative Phase Names for Ground Segments
Phase 0	Concept Studies	
Phase A	Concept Development	
Phase B	Preliminary Design	
Phase C	Complete Design	
Phase D1	Fabrication and Integration	Implementation and Integration
Phase D2	Fielding and Checkout	Deployment
Phase D3	Operations and Support	

The DOD acquisition lifecycle phases from [3] aligned with respect to the MAG phases from [4] are shown in Figure 14-2. This figure also shows the usual positioning of the major technical reviews [system/segment requirements review (SRR), system design review (SDR)/system functional review (SFR), software requirements and architecture review (SAR), preliminary design review (PDR), critical design review (CDR), and test readiness review (TRR)] that are most applicable to ground segment development. To understand how the information in this chapter that relates to the DOD acquisition lifecycle phases can be applied to another acquisition lifecycle model, align that lifecycle models' major technical reviews and acquisition milestones (gates for passing from one acquisition lifecycle phase to the next) with those shown in Figure 14-2.

Specific acquisition lifecycle models most applicable to ground segment acquisition are discussed in section 14.4, and specific contractor development lifecycle models most applicable to ground segment development are discussed in section 14.5. Technical considerations and lessons learned that apply to the acquisition lifecycle in general are discussed in the following subsections.

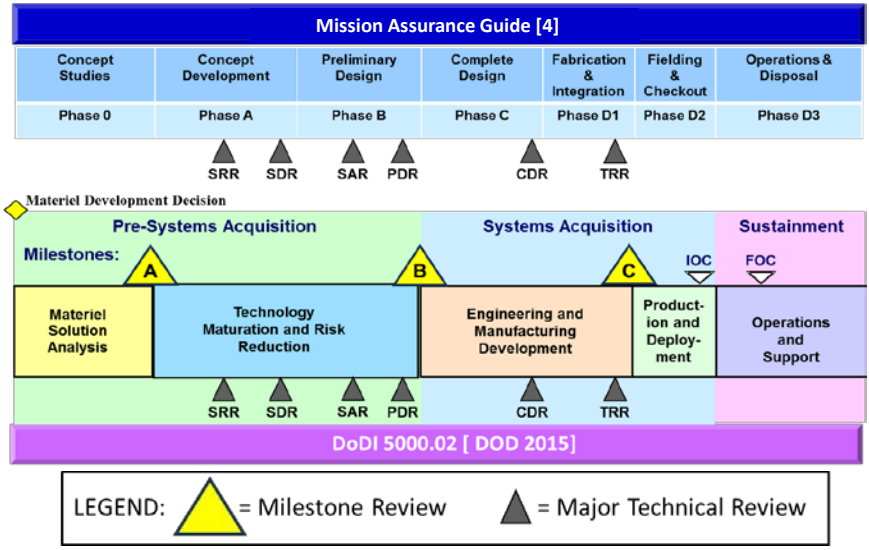


Figure 14-2. MAG and DOD lifecycle phases aligned.

14.3.1 Overall Acquisition Lifecycle Technical Considerations

During the early acquisition lifecycle phases, the acquisition team focuses on the systems engineering needed to define the system to be acquired. As time proceeds across the acquisition phases, the focus changes to monitoring the contractor team's engineering of the system they are developing. Across the acquisition phases, however, the acquisition team is responsible for producing a number of systems engineering products. Table 14-2 provides a list of key products that must be produced by the user organization and acquisition team during each acquisition phase. The use of an "I" in a table column in Table 14-2 indicates that initial preparation of the product, while the use of a "U" in a table column indicates the product is updated.

During the Materiel Solution Analysis (MSA) phase, the emphasis is on determining the optimal solution for satisfying a documented need. The acquisition team may have a number of contractors in this phase working on system architecture studies, technology maturation, and similar topics. By the end of the phase, the acquisition team will have chosen the preferred system architecture, determined the contractual requirements for the system to be acquired (documented in a TRD or SRD, prepared the RFP for the next acquisition phase, performed the initial program planning, and prepared for the acquisition gate for entering the next phase (Milestone A).

Table 14-2 Key Acquisition Systems Engineering Products

Type of Product	DOD Name of Product	Applicable DOD Acquisition Phase				
		MSA	TMRR	EMD	PD	OS
User requirements	Initial Capabilities Document (ICD)	I				
	Capability Development Document (CDD)	I	U	U		
	Capability Production Document (CPD)			I	U	
User concept of operations	Concept of Operations (CONOPS)	I	U	U	U	U
System requirements	Technical Requirements Document (TRD) [(or System Requirements Document [SRD])	I	U	U	U	U
Analysis of alternative materiel solutions	Analysis of Alternatives (AoA)	I				
System architecture	Preferred System Architecture	I				
	Government Reference Architecture (GRA)		I	U		
Threat assessment	System Threat Assessment Report (STAR)	I	U	U	U	U
Risk assessment	Risk Assessment	I	U	U	U	U
Technology readiness	Technology Development Strategy (TDS)	I				
	Critical Technology Elements (CTEs)	I				
	Technology Readiness Assessment (TRA)		I	U		
Test and evaluation planning	Test and Evaluation Strategy (TES)	I				
	Test and Evaluation Master Plan (TEMP)	I	U	U	U	
Acquisition strategy planning	Acquisition Strategy Plan (ASP)	I	U	U	U	U

I = initial preparation; U = update; MSA = mission solution analysis; TMRR = technology maturation and risk reduction; EMD = engineering and manufacturing development; PD = production and development; OS = operation and support

Type of Product	DOD Name of Product	Applicable DOD Acquisition Phase				
		MSA	TMRR	EMD	PD	OS
Intellectual property planning	Intellectual Property (IP) Plan (documented as part of the ASP)	I	U	U	U	U
Systems engineering planning	Systems Engineering Plan (SEP)	I	U	U	U	U
Sustainment planning	Life Cycle Sustainment Plan (LCSP)	I	U	U	U	U
Information support planning	Information Support Plan (ISP)	I	U	U	U	U
Program security planning	Program Protection Plan (PPP)	I	U	U	U	U
Cybersecurity strategy	Cybersecurity Plan (documented as an appendix to the PPP)	I	U	U	U	U
Software assurance strategy	Software Assurance Plan (documented as an appendix to the PPP)	I	U	U	U	U
Program safety evaluation	Programmatic Environment, Safety, and Occupational Health Evaluation (PESHE)	I	U	U	U	U
Software acquisition planning	Software Acquisition Management Plan (SWAMP)	I	U	U	U	U
Clinger Cohen Act (CCA) Compliance	Clinger Cohen Act Compliance Assessment	I	U	U	U	U
Cost analysis requirements	Cost Analysis Requirements Document (CARD)	I	U	U	U	
Request for proposals	Requests for Proposal (RFPs)	I	I	I	I	I
Acquisition milestone/ decision point material	Acquisition Milestone Documentation or Acquisition Decision Point Documentation	I	I	I	I	

I = initial preparation; U = update; MSA = mission solution analysis; TMRR = technology maturation and risk reduction; EMD = engineering and manufacturing development; PD = production and development; OS = operation and support

During the Technology Maturation and Risk Reduction (TMRR) phase, the acquisition team is still focused on defining the system that is to be developed. There may be multiple competing contractors in this phase, each performing technology maturation, system requirements analysis, system architectural design, and preliminary design. The acquisition team must review the work of the TMRR contractors and update the contractual requirements for the system that will be developed in the next phase, taking into account the results of the contractors in this phase. By the end of the TMRR phase, the acquisition team will have updated the system requirements in the TRD, prepared the RFP for the next acquisition phase, updated the program planning, and prepared the acquisition gate for entering the next phase (Milestone B).

During the Engineering Manufacturing and Development (EMD) phase, the contractor team will design, develop, integrate, and test the system to meet the contractual requirements. Generally only one contractor will be selected to develop the system. The acquisition team's focus during this phase is to review the development contractor's work to ensure that the system is being developed correctly. By the end of this phase, the acquisition team will have updated the contractual requirements for the system and its planning for the next acquisition phase, prepared the RFP for the next phase (if needed), and prepared for the acquisition gate for entering the next phase (Milestone C). Passing this acquisition gate gives the acquisition team permission to proceed with production and deployment.

During the Production and Deployment phase (PD), the contractors will deploy the system developed during EMD. This may include launching satellites (if the development contract was for a complete space system), deploying the ground segment at the operational site(s), or both. The user organization will perform operational test and evaluation and, if the system passes, will begin initial operations. By the end of this phase, full production and/or deployment will be complete and the system will enter its full operational capability (FOC). During this phase, the acquisition team is focused upon ensuring that the production, deployment, and operational testing proceed smoothly. There may be a new contract for this phase, especially if production of system components must be performed. However, for a system without production, the EMD contract may include deployment as well as development.

The acquisition team has a secondary focus during the PD phase, and that is to prepare for the Operation and Support (OS) phase. When the system transitions to sustainment, the acquisition team usually passes management of the system acquisition to a logistics team who oversees the hardware and software sustainment throughout the operational life of the system. During the PD phase, the acquisition team usually works with the logistics team to update the planning necessary for sustainment and to prepare the RFP(s) for one or more sustainment contracts. The logistics team is responsible for ensuring the system remains in

an operational condition (i.e., for ensuring the hardware and software maintenance is performed efficiently and effectively) and for ensuring the system is enhanced to meet new requirements as they are identified.

14.3.2 Overall Acquisition Lifecycle Lessons Learned

If the ground segment is the entire system being acquired, then the focus of all of the acquisition lifecycle phases is on the ground segment. However, in a system where multiple segments are being acquired, only one of which is the ground segment, the ground segment is frequently ignored in the early acquisition phases. The acquisition team must ensure that the early systems engineering addresses the ground segment as well as the space, launch, or user segments. The ground segment development and deployment is usually along the critical path during EMD and PD due to the size and complexity of its software. The ground segment is an important part of the system architecture and must be included in architecture studies, technology maturation, and system requirements development. In addition, the ground segment is frequently critical to the mission and must be in place to support launch and mission operations.

The acquisition team must satisfy many stakeholders throughout the acquisition lifecycle. These stakeholders include the acquisition team's higher-level acquisition authorities, the operators of the ground segment site(s), the users of the system's products who may come from different organizations responsible for diverse missions, and personnel from organizations responsible for interfacing segments and systems. The stakeholders may have conflicting needs, and usually have different priorities for various capabilities of the ground segment. The acquisition team is responsible for performing the technical, cost, and schedule tradeoffs among the stakeholder's needs, for making decisions among conflicting user needs, for establishing priorities among ground segment requirements, for developing the RFPs that result in the contractor team(s) creating high-quality products that meet the ground segment requirements, and for ensuring that the ground segment under development will meet its cost and schedule constraints.

Ground segment acquisition usually involves the development and integration of large, complex, mission critical software. Information on preparing RFPs for such a system can be found in [5].

14.4 Ground Segment Acquisition Lifecycles

The development of ground segments generally involves implementation of computer-related, communications-related, and security-related hardware and installation of that hardware in one or more operational sites around the world. In addition, most ground segments contain a large amount of software, including newly developed, legacy reuse, modified reuse, commercial off-the-shelf, and

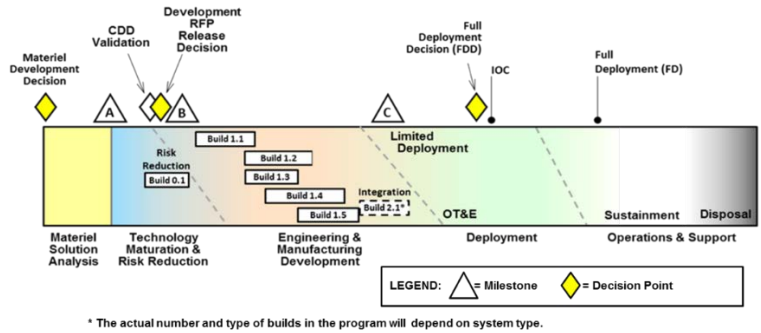
sometimes open source software. Because of the size and complexity of ground segments, the ground segment itself and the software within it are usually developed in an iterative manner, in a series of builds.

While Figure 14-2 shows the acquisition lifecycle in a "once-through" or waterfall lifecycle, this is not the optimal method of acquisition for ground segments. Even at the acquisition level, consideration of ground segment iterations and structuring contracts accordingly is advisable for large, complex ground segments. The DOD has recently issued an update to DODI 5000.02 [3] which defines acquisition lifecycle models that are more appropriate for ground segments. Other acquisition organizations may have concepts similar to those in [3].

There are six acquisition lifecycle models in [3]. The goal is for each program to select the model that is closest to their situation and to tailor it for their program. The models include:

Model 1: Hardware Intensive Program
Model 2: Defense Unique Software Intensive Program
Model 3: Incrementally fielded software intensive program
Hybrid Program A (Hardware Dominant)
Hybrid Program B (Software Dominant)
Model 4: Accelerated Acquisition Program

The two most appropriate models for ground segments are Model 2 and Model 3, which are shown below in Figures 14-3 and 14-4. For ground segments being acquired as part of a space or launch system, where new space or launch hardware development and possibly production are coupled with the ground segment development, one of the two hybrid models will probably be more appropriate. This chapter discusses only Models 2 and 3 as examples. The two corresponding hybrid models are very similar.

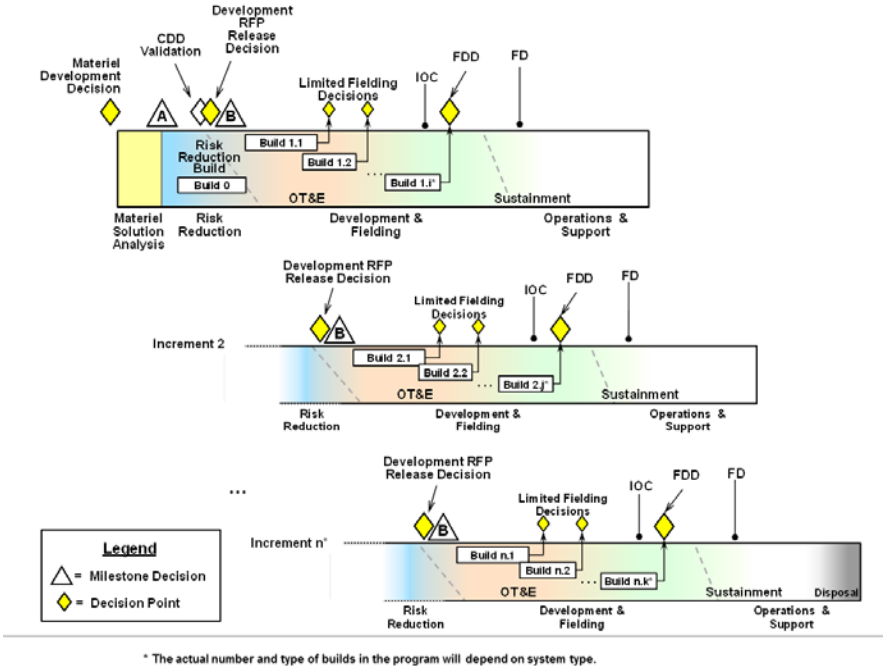


* The actual number and type of builds in the program will depend on system type.

IOC = initial operational capability, OT&E = operational test and evaluation

Figure 14-3. Model 2: Defense unique software intensive program [3].

In Model 2, the acquisition lifecycle is shown as a "once-through" set of acquisition phases (see Figure 14-3). However, the TMRR and EMD phases show a series of software development builds. This model is useful for a ground segment dominated by the need to develop complex, usually unprecedented, software that will not be fully developed until a number of software builds have been completed.



* The actual number and type of builds in the program will depend on system type.

Figure 14-4. Model 3: Incrementally fielded software intensive program [3].

In Model 3 (see Figure 14-4), the acquisition lifecycle is shown as multiple copies of the same model as in Figure 14-3, each beginning at a later point in time and labeled as a new increment of the system being acquired. A later increment may provide additional ground segment capabilities deployed into operations. Such increments are frequently called block upgrades. Alternatively, a later increment may be a technology update increment that incorporates new technology into the ground segment, including refreshing the commercial off-the-shelf (COTS) hardware and software, but does not provide new capabilities. Most ground segment block upgrades incorporate new technology in addition to providing new capabilities. As in Model 2, each increment requires a series of software builds in order to develop the complex software needed to meet the increment requirements or to incorporate the new technology.

Upgrades to existing ground segments usually use an acquisition lifecycle model adapted to the extent and risk of the upgrade. The full acquisition lifecycle would be applied, for example, if the upgrade requires extensive system architecture changes and new technology that needed maturing. Usually, however, upgrades to existing ground segments do not require repetition of the MSA phase. Frequently upgrades use a single contract for system requirements, preliminary and complete design, implementation, and deployment (i.e., the Technology Maturation and Risk Reduction, EMD, and PD phases). Because upgrades will require iteration in software development and possibly in ground segment development, builds and increments will be needed. Thus, upgrades to existing ground segments will use a version of Model 2 or Model 3 adapted to the needs of the upgrade.

The acquisition team is responsible for planning the acquisition system development, selecting and adapting one of the lifecycle models; producing the RFPs; selecting the contractor team(s) that will perform the contract; and overseeing the contractor team's performance, adherence to the contractor team's documented processes, and the quality of the contractor team's products.

14.4.1 Ground Segment Acquisition Lifecycles Technical Considerations

The effective use of these acquisition lifecycles requires an in-depth technical understanding of the ground segment to be acquired. Based on contractor and government architecture studies performed during the MSA phase and on the ground segment architectures developed by contractors during the TMRR phase, the government must develop a GRA that they can use for preparing the ground segment requirements, the cost and schedule estimates for the development, and the RFPs for the TMRR and EMD/PD phases. In addition, when using Model 3, the acquisition team must understand the ground segment architecture enough to structure the increments so that each increment builds architecturally on the preceding increment.

When using Model 3, the acquisition team must determine the set of ground segment requirements to be implemented in each increment. This necessitates understanding the requirements in sufficient depth to be able to prioritize the requirements and allocate them to the acquisition increments. In addition, the acquisition team must understand the dependencies among the requirements so that when a requirement is assigned to an increment, all requirements upon which that requirement depends are assigned to the same or preceding increments.

The acquisition team must also understand the ground segment well enough to develop a feasible contract schedule. For most ground segment acquisitions, the ground software development is the schedule driver due to the software's size

and complexity. This means that the acquisition team must understand the software well enough that an accurate estimate of the software schedule can be made. For Model 3, the acquisition team must be able to estimate accurately the schedule of each increment of the ground segment.

14.4.2 Ground Segment Acquisition Lifecycles Lessons Learned

When using acquisition lifecycle Model 3, the acquisition team not only must determine the set of requirements allocated to each increment, but also must determine how much to overlap the increments. The increments should be enough separated in time that the lessons learned from initial operations of earlier increments can be incorporated into subsequent increments. In addition, the TMRR phases for the increments should not overlap because this would mean that a later increment would be developed on an unstable requirements and architecture baseline from the preceding increment. Similarly, the development integration and testing of an increment in the EMD and PD phases should be complete before the subsequent increment is integrated with it and the combination tested. Otherwise, the subsequent increment will be integrated with an unstable product baseline from the preceding increment that will likely contain undiscovered defects because it is not fully tested. Thus, the development integration and testing of the previous increment should be completed before the subsequent increment begins to be integrated and tested with it. Finally, the transition of consecutive increments into operations must be well coordinated and not cause operational conflicts since, depending on the amount consecutive increments overlap, initial deployment of the later increment may conflict with full deployment of the earlier increment.

The DOD acquisition lifecycle figures in DOD instruction 5000.02 [3] do not include the major technical reviews. This allows the acquisition team to specify in the contract when these reviews will take place. Most large ground segments are developed in increments, whether the contract requires multiple increments delivered to operations or the contractor proposes to develop, integrate, and test the ground segment in multiple increments for risk reduction. When increments are used, the entire design is not complete until the final increment is designed. Requiring a series of fixed waterfall reviews, that is, one SAR, PDR, and CDR on a fixed schedule is inconsistent with developing the ground segment in multiple increments. The contract should be written to require the SAR, PDR, and CDR reviews for each increment, and to require the contractor to schedule these reviews appropriately. The acquisition strategy needs to address this approach up front so that the entire acquisition management chain understands the acquisition team's rationale and agrees to this approach.

For new DOD systems with a TMRR phase, a SAR and PDR are usually held during the TMRR phase while the CDR is held during the EMD phase. However, when the contractual requirements (e.g., in the TRD) change between

the TMRR and EMD phases, the EMD contract should specify that the contractors hold delta SAR(s) and PDR(s) to address the changes. Thus, for a ground segment developed in multiple increments, a delta SAR and delta PDR should be held for each increment. Full SARs and PDRs should be held if the requirements changes are significant enough to cause architectural changes or major changes to hardware or software items.

In spite of the fact that software builds are shown in DOD acquisition lifecycle models 2 and 3, the acquisition team should not specify the content or schedule of the software builds, the software development lifecycle, or the software methodologies that the contractors must use. The contractor should always have the freedom to determine how the software will be developed, based on their organizational software processes. If there is a need for early releases of the ground segment to operations, the contract should specify only the schedule and requirements for these releases.

14.5 Ground Segment Development Lifecycles

The ground segment contractor team must define their development lifecycle within the constraints of the acquisition lifecycle model used and the contract requirements, including the required ground segment delivery (or deliveries) and schedule. The principal development lifecycle phases are shown in Table 14-2, aligned with the corresponding MAG and DOD acquisition lifecycle phases. As Table 14-2 shows, the development lifecycle phases do not have a one-to-one correspondence with either the MAG phases or the DOD acquisition lifecycle phases.

The development lifecycle phase names shown in the first column are those used in this handbook. These names are not consistent across the industry, so contractors may use different phase names. However, the principal development activities performed in each phase should be consistent no matter what name is used for the development phase (see Section 14.6).

Table 14-3. Development Lifecycle Phases

Development Lifecycle Phase	Corresponding MAG Phase(s)	Corresponding DOD Acquisition Lifecycle Phase(s)
Concept Development	Concept Studies	Materiel Solution Analysis
Requirements	Concept Development	Technology Maturation and Risk Reduction
Architectural and Detailed Design	Preliminary Design Complete Design	Technology Maturation and Risk Reduction

Development Lifecycle Phase	Corresponding MAG Phase(s)	Corresponding DOD Acquisition Lifecycle Phase(s)
		Engineering and Manufacturing Development
Product Development	Implementation and Integration	Engineering and Manufacturing Development
Test Planning and Execution	Implementation and Integration	Engineering and Manufacturing Development Deployment
Transition to Operations	Deployment	Deployment Operations and Support
Maintenance and Support	Operations and Support	Operations and Support

Figure 14-5 depicts an example alignment of development lifecycle phases with acquisition lifecycle phases and major technical reviews. In this example, some development lifecycle phases are executed concurrently, and some overlap. It is almost never appropriate to define a strictly sequential development lifecycle model, especially if the phases do not overlap. For example, in the TMRR phase, the ground segment architecture is developed concurrently with the ground segment requirements. This reflects the need for architecture and requirements development to be iterative with each other and ensures that the requirements and architecture are consistent.

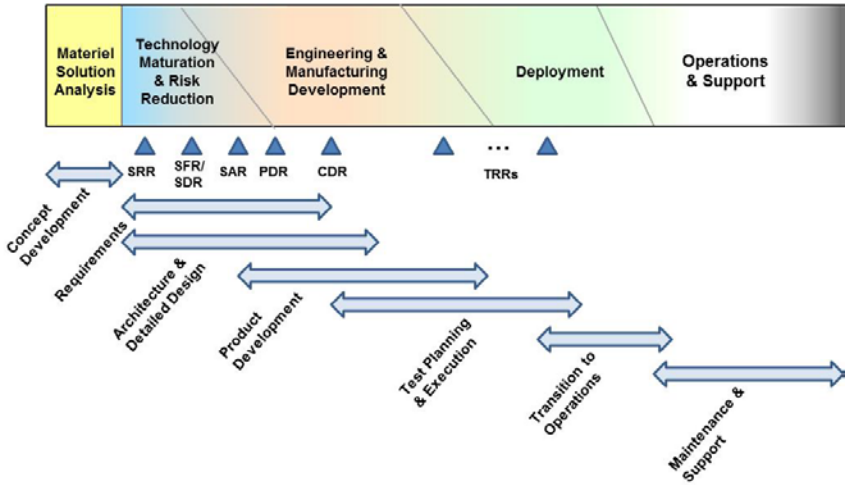


Figure 14-5. Example alignment of acquisition and development lifecycle phases.

Even though a once-through acquisition lifecycle model is used (e.g., Model 2), the development contractor team may decide that the risk of using this development lifecycle model is too large, especially for large ground segments. In this case, even if the acquisition organization has not defined increments to be deployed into operations, the development contractor team may use an iterative development lifecycle model, defining internal ground segment increments to be designed, implemented, integrated and tested, and possibly deployed, but not necessarily transitioned to operations. Figure 14-6 depicts an example of multiple development increments aligned with the acquisition phases and major technical reviews.

Figure 14-6 shows three development increments, following a concept development phase aligned with the MSA acquisition phase and the ground segment requirements, architecture, and preliminary design phases aligned with the TMRR acquisition phase. The first two development increments are internal increments; that is, the only increment to transition to operations; maintenance is the third and final increment. Thus, Figure 14-6 show the transition to operations and maintenance and support development phases occurring once following development of the third increment. If the contract specifies incremental deliveries of the ground segment to operations, then the increments corresponding to the contractual deliveries will be followed by transition to operations.

The TMRR acquisition phase generally includes an SRR, SDR/SFR, SAR, and PDR at the ground segment level. Each increment shown in Figure 14-6 includes all of the development phases from requirements through test planning and execution for that increment, and each has a SAR, a PDR, and a CDR specific to that increment. Each increment also has TRRs for the testing of the requirements for that increment. Each increment builds on the preceding increment, so the third increment comprises the entire ground segment. The test planning and execution phase for the third increment includes development testing of the entire ground segment.

There are a great many ways that the development contractor can define their development lifecycle. Figures 14-5 and 14-6 are just two examples. The acquisition team must understand the contractors development lifecycle and ensure that it is feasible, will enable the contract requirements to be met, and will produce a high-quality ground segment development.

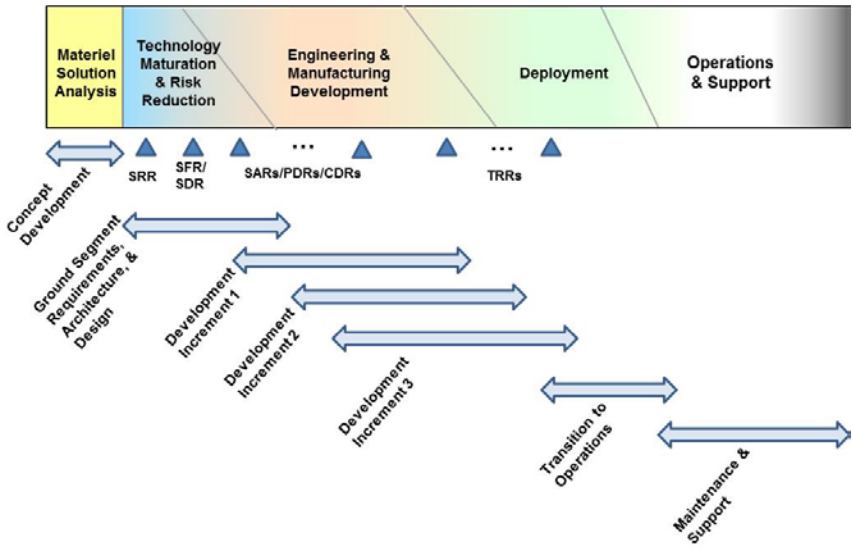


Figure 14-6. Example of multiple development increments.

14.5.1 Ground Segment Development Lifecycle Technical Considerations

Whether the acquisition team requires incremental deliveries in the contract or the development contractor team defines implementation increments with only the final increment transitioning into operations, the acquisition team must thoroughly understand the contents of the increments. The definition of the increments is a complex systems engineering task that involves the allocation of ground segments requirements and architecture elements to the increments. The acquisition and development teams must ensure that the contents of the increments satisfy the following:

- Functional capabilities
  - The architecture elements needed to implement the requirements allocated to an increment must be assigned to that increment or to an earlier increment.
  - The early increments must create the necessary hardware and software infrastructure or building blocks to support the needs of the later increments.
  - Each increment must define end-to-end threads of functional capability that can be exercised to verify the requirements allocated to the increment, whether or not the increment is transitioned into operations.

    - Each increment must provide increasing ground segment functionality over the previous increment.
- Schedule considerations
    - The increments must support the ground segment simulator, testbed, and test facility development by providing the necessary interfaces when needed.
    - The increments must support the overall ground segment test schedule by providing the necessary capabilities when needed.
    - The increments must meet the contractual requirements and schedule for increments to be transitioned to operations.
    - The schedule for each increment must be feasible, given the requirements and architecture elements that must be developed, and schedule margin should exist for each increment.
    - The overlap of the increments must also be feasible.
- Resource constraints
    - The implementation of the increments must meet the budget limitations, for example, fiscal year dollar constraints.
    - The implementation of the increments must meet manpower limitations.
    - The implementation of the increments must balance the workload of the various disciplines (e.g., computer hardware, communications, security, algorithm development, software infrastructure, databases, and user interfaces) across the increments.

Whether the contractor develops the ground segment in increments or not, the software will most likely be developed in a series of builds using an iterative type of software development lifecycle. The development contractor will determine the software development lifecycle and methodologies to be used, plus the number and content of the software builds.

14.5.2 Ground Segment Development Lifecycles Lessons Learned

The acquisition team should always beware of the contractor team proposing a strictly waterfall development lifecycle model with no overlap of lifecycle phases. Requirements specification and architecture development are best accomplished in parallel because there must be iteration between these activities. Test planning always starts before product development is completed, usually as early as CDR, to have enough time to prepare for and execute the test program. Product development of the individual software items begins with the SDR/SFR. Similarly, hardware development also begins at this time, especially if the hardware includes new technology. The acquisition team needs to ensure the contractor team understands this and adjusts their development lifecycle phases accordingly.

The implementation of ground segment hardware occurs concurrently with the ground segment software development. The hardware implementation and software build content must be coordinated so that effective integration and testing of the hardware and software can occur as the increment development proceeds. The acquisition team must be sufficiently knowledgeable about the hardware implementation and software build content to be able to determine whether the contractor's integration and test plans are feasible. This is true whether or not the contractor develops the ground segment in increments.

When the ground segment is developed in increments, the development team not only must determine the content of each increment, but also must determine how much to overlap the increments. The increments should be separated enough in time that the lessons learned from earlier increments can be incorporated into subsequent increments. In addition, the requirements and architecture phases for subsequent increments should not overlap, because this would mean that a later increment would be developed on an unstable requirements and architecture baseline from the preceding increment. Similarly, integration and testing of an increment should be complete before the subsequent increment is integrated with it and the combination tested. Otherwise, the subsequent increment will be integrated with an unstable product baseline from the preceding increment that will likely contain undiscovered defects since it is not fully tested. The acquisition team must be able to determine whether the overlap of the increments is feasible.

Finally, while concurrency of the increments is usually necessary to meet contract schedules, too much concurrency can make the ground segment development unexecutable. If there are too many increments that integrate with one another planned to be under development concurrently, the management of the increments is extremely difficult to execute, especially the concurrent manpower needs of the increments under development. A general rule of thumb to use is that no more than three increments that integrate with each other should be under development concurrently. If the contractor team proposes more than three increments under development concurrently, the acquisition team must obtain from the contractor team justification that such concurrence is necessary and evidence of successful prior execution of such a plan.

14.6 Ground Segment Development Lifecycle Phases

This section discusses the activities performed in each of the contractor's ground segment development lifecycle phases (see Table 14-3). Each development contractor team is responsible for planning the ground segment development; specifying their team's development lifecycle model(s); planning their development phases, activities, and processes to be used; and performing their planned processes. The acquisition team has an oversight role in each of these activities, ensuring that the contractors adhere to their processes, perform to the

planned schedule, and produce products of acceptable quality that satisfy their requirements.

Most contractors have organizational processes and procedures for systems engineering and software engineering. The systems engineering processes and procedures usually align with the CMMI® for Development [2] and with a systems engineering standard such as *IEEE 15288, Systems and software engineering – System life cycle processes* [6]. The software engineering processes usually align with the *Software Development Standard for Mission Critical Systems* [1].

This section is written with the assumption that the contract is solely for a ground segment (or part of a ground segment) rather than for a system one of whose segments is a ground segment. The discussion can be easily adapted for the case of acquiring a system one of whose segments is a ground segment.

There are many other tasks the development contractor must perform in addition to the basic activities in each of the development lifecycle phases. Specialty engineering activities (e.g., reliability, maintainability, and availability; safety; security; and human systems integration) occur throughout the development. Other activities that occur throughout the development lifecycle include risk assessment and management, configuration and data management, quality assurance, and metrics.

14.6.1 Concept Development Phase

The concept development phase is an early lifecycle phase whose objective is to define the preferred ground segment architecture and develop the ground segment contractual requirements. Most of the products to be produced during this phase are the responsibility of the acquisition team (see Section 14.3 and Table 14-2). For DOD systems, the concept development phase aligns with the MSA acquisition phase.

During the concept development phase, contractors perform ground segment trade studies and architecture studies, and provide input to the acquisition team's analysis of alternative architectures and definition of the preferred ground segment architecture for the defined need. Contractors also contribute to maturing technologies that will be needed for the ground segment development and contribute to the ground segment contractual requirements by providing input to performance requirements based on the results of their technology studies and prototype testing. The products produced by the contractors during this phase are mostly study results and new technology hardware and software prototypes. The key acquisition products for this phase are given in Table 14-2.

There are many stakeholders for the concept development phase, including the acquisition team, higher level acquisition management organizations, users of the system to be acquired, owners of interfacing systems, test and evaluation personnel, sustainment personnel, security personnel, contractors interested in developing the system or part of the system being acquired, and contractors under contract during the MSA phase.

14.6.2 Requirements Phase

The requirements phase begins during the MSA acquisition phase and continues through the TMRR phase and into the EMD phase. The requirements phase consists of that period of time when a principal activity of the acquisition team or contractor team is requirements engineering to develop the requirements baseline. Requirements engineering begins with the acquisition team who is responsible for creating and updating the ground segment TRD. The TRD becomes the contractual requirements document used in the RFPs for the TMRR, EMD, and PD acquisition phases.

The requirements phase for the contractor team spans the TMRR and EMD acquisition phases. If contractual requirements change for the PD phase, the requirements phase will extend into that acquisition phase. Requirements engineering includes requirements elicitation, analysis, definition, documentation, allocation, elaboration, derivation, validation, and management. The contractor team prepares the ground segment System Specification (SS) based on the ground segment TRD and their ground segment architecture. This specification is reviewed at the SRR in the TMRR phase. The acquisition team must perform a thorough review of the ground segment requirements documented in the ground segment SS to ensure that the ground segment requirements fully cover all of the acquisition team's requirements in the TRD. The ground segment SS frequently replaces the TRD as the contractual set of requirements after the acquisition team has approved the contractor team's SS.

Based on the ground segment architecture, the ground segment requirements are allocated to ground segment subsystems. The subsystem requirements are then elaborated, derived, and documented in subsystem requirements specifications. The subsystem requirements specifications and the ground segment architecture are reviewed at the SDR/SFR in the TMRR phase. For each subsystem, the architecture defines the subsystem's components and the hardware and software items for the components. The subsystem requirements are allocated to components, and component requirements are elaborated, derived, and documented in component requirements specifications. The component requirements are then allocated to hardware and software items. The requirements for the hardware and software items are elaborated, derived, and documented. The component and hardware and software item requirements are

reviewed along with the ground segment preliminary design at the PDR in the TMRR phase.

Requirements definition in the EMD phase consists of updating and finalizing the full requirements baseline per any changes in the TRD requirements, ground segment architecture, or preliminary design and per impacts of the detailed design. The full requirements baseline is provided for review at the CDR in the EMD phase. It is essential that the acquisition team provide a thorough review of the requirements at all levels to ensure the ground segment hardware and software items, components, and subsystems are being built to a correct and complete set of high quality requirements.

Throughout the requirements definition phase, it is essential that the requirements be documented and kept up to date. Bi-directional traceability between the levels of requirements is important to be created and maintained as well. Such traceability provides assurance that all TRD and ground segment requirements are properly covered by the lower level requirements. The bi-directional traceability also provides the capability to determine the impact of requirements changes on the ground segment subsystems, components, and hardware and software items. Maintenance of the requirements baseline and the bi-directional traceability is called requirements management. Requirements management must be performed by the contractor throughout the development lifecycle. There are numerous commercial tools available that can assist with this activity.

There are many government stakeholders for the requirements phase, including the acquisition team, higher level acquisition management organizations, users of the system to be acquired, logistics organizations, specialty engineering organizations, owners of interfacing systems, test and evaluation personnel, sustainment personnel, and security personnel. There are also numerous contractor stakeholder organizations, including the contractors under contract in the TMRR and EMD phases and their subcontractors, and various organizations within the contractors and subcontractors, such as systems engineering, integration and test, product development organizations, specialty engineering organizations, and logistics organizations.

14.6.3 Architecture and Detailed Design

The architecture and detailed design phase begins during the MSA acquisition phase and continues through the TMRR phase and into the EMD phase. The architecture and detailed design phase consists of that period of time when a principal activity of the acquisition team or contractor team is developing the architecture, the preliminary design, and the detailed design of the ground segment as a whole and of its constituent subsystems and hardware and software items.

Architecture and detailed design begins with the acquisition team who is responsible for developing the preferred ground segment architecture. This architecture is a high level architecture, with only enough detail to guide the acquisition team in planning the acquisition, developing the requirements and RFP for the next phase, and estimating the system cost and schedule. The architecture is documented in the AoA and the preferred architecture is provided in the ASP (see Table 14-2).

Contractors may begin the architecture and detailed design phase during the MSA acquisition phase if they are on contract to perform ground segment architecture studies. During the TMRR acquisition phase, the contractor team develops the ground segment architecture and design, which is reviewed at the SDR/SFR. The ground segment architecture and design is usually documented in a system/segment design document (SSDD), supported by trade studies, engineering analyses, and prototypes. The ground segment architecture and design decomposes the ground segment into subsystems and then into components and hardware and software items. The subsystem requirements are also usually reviewed at the SDR/SFR.

Also during the TMRR acquisition phase, the contractor team develops the architecture and requirements of the software items, which are reviewed at a SAR, and develops the full preliminary designs of the software items. The requirements and preliminary designs of the components and hardware items are similarly prepared. The component, hardware, and software requirements are documented in requirements specifications. The software architecture is documented in a software architecture description, and the software preliminary design is documented in software design documentation. The hardware preliminary design is usually documented in drawings and hardware design documentation. The architecture and preliminary design of the components are documented appropriately, based on whether the components contain both hardware and software items, only hardware items, or only software items. The preliminary design of the ground segment, its subsystems, components, and hardware and software items are reviewed at the PDR.

During the EMD phase, the contractor team updates the component and hardware and software item requirements, architectures, and preliminary designs developed in the preceding phase, based on any changes in the contractual requirements. These may be reviewed at one or more delta SARs and PDRs if the changes are significant. During EMD, the contractor team also performs the detailed designs of the hardware and software items, which are reviewed at one or more CDRs. If the ground segment is developed in increments, these reviews (SARs, PDR, and CDRs) should be held for each increment since the detailed designs of the hardware and software items, as well as the requirements, architecture and preliminary design changes, will be developed for each increment, building upon the preceding increment. Thus, when the ground

segment is developed in increments, the architecture and detailed design phase proceeds through the detailed design of the last increment.

It is essential that the acquisition team provide a thorough review of the architecture, preliminary designs, and detailed designs of the ground segment, subsystems, components and hardware and software items to ensure the ground segment will meet its contractual requirements and will properly perform its functions when developed.

There are many government and contractor stakeholders for the architecture and design phases, similar to those listed above for the requirements phase.

14.6.4 Product Development

The product development phase is performed during EMD to develop, integrate and test the hardware and software items. The hardware and software items are also integrated into components and subsystems and the results tested during the product development phase. This phase is executed solely by the contractor team, with the acquisition team in an oversight role.

At the current time, most hardware items, or their constituent units, are commercial off-the-shelf (COTS) products, rather than products that are uniquely developed by the contractor team. Thus the hardware product development includes procuring the appropriate COTS products based on the detailed design of the hardware items and integrating and testing those products into the hardware units and items. Hardware product development includes development testing to ensure the hardware units and items under construction satisfy their designs. Hardware product development also includes testing of the hardware units to verify they meet their documented requirements. If the ground segment is being developed in increments, the hardware items will usually be developed for the first increment where they are needed and possibly updated in future increments to add necessary capability.

The software for ground segments may include newly developed software, modified or unmodified reuse software, open source software, and/or COTS software. Whether or not the ground segment is developed in increments, the software will almost always be developed using some type of iterative lifecycle model (e.g., incremental, evolutionary, spiral, and agile), where the software requirements definition, architecture and design, implementation, integration, development testing, and qualification testing are performed in an iterative fashion. Depending upon the software lifecycle model, those software activities may be performed concurrently and in any order. Product development for software includes integration of all of the types of software (new, reuse, open source, and COTS) into software items. Development testing of the newly developed and modified reuse software and of the integrated software is

performed during product development to ensure the software satisfies its design. Software qualification testing is also performed to verify that the software items meet their documented requirements.

Product development also includes the integration of hardware and software items into components and subsystems, per the detailed design. This includes tasks to integrate hardware items together and to load software items into the target hardware. It also includes tasks to test the integrated hardware and software items to ensure the integrated components and subsystems function as expected per their designs (development testing) and meet their requirements (verification). Such testing includes all test methods including analysis (e.g., for verification of the specialty engineering requirements allocated to the subsystems and components) and inspection (e.g., for required hardware and software characteristics that can be verified by visual inspection) as well as demonstration and test.

Each hardware item and each software item should have its requirements, architecture, and design reviewed at requirements, architecture and design reviews (i.e., SRR, SAR, PDR, CDR). In addition, testing of each hardware item and each software item to ensure its requirements are met should be preceded by a test readiness review (TRR).

There are many stakeholders for the product development phase. Government stakeholders include the acquisition team, users of the system to be acquired, logistics organizations, specialty engineering organizations, owners of interfacing systems, test and evaluation personnel, sustainment personnel, and security personnel. There are also numerous contractor stakeholder organizations, including the contractors under contract in the EMD phase and their subcontractors, and various organizations within the contractors and subcontractors, such as systems engineering, integration and test, product development, specialty engineering, logistics, and sustainment organizations.

14.6.5 Test Planning and Execution

The test planning and execution phase is executed during the Engineering and Manufacturing Development and Deployment phases. The test planning and execution phase first includes integrating the ground segment subsystems into the full ground segment and testing the interfaces among the subsystems to ensure that the subsystems function as designed and that all of the subsystem requirements are verified in the integrated environment. Following this, verification testing of the integrated ground segment is performed to verify that the ground segment meets all of the requirements as documented in the ground segment specification. Verification testing includes using all test methods, as specified in the ground segment requirements specification. Finally, end-to-end testing is performed of the ground segment in its full integrated environment,

including the entire enterprise with all interfacing systems and personnel. The end-to-end testing usually completes the verification of ground segment requirements that had previously only been tested with simulated interfaces. Following full verification of the ground segment requirements, the acquisition team is able to consider the contractual ground segment requirements as "sold off".

Also included in the test planning and execution phase is operational testing of the completed ground segment. Operational testing may be performed by an independent government operational test and evaluation (OT&E) organization or by the user organization, depending upon the acquisition environment. Operational testing is usually performed in the operational environment with operations personnel executing the functions, instead of development contractor personnel, and using actual operational procedures and operational databases. Successful operation testing of the ground segment is usually necessary for the user organization to accept the system from the acquisition organization.

All of this testing requires preparation of test plans, test cases, and test procedures. The test procedures need to include the expected results and acceptable deviation from the expected results for the test step(s) to be considered successful. Dry runs of the test procedures should be held and any problems should be corrected before the formal testing is held. Preparation for testing is a complex and time-consuming process. The acquisition team must ensure the test schedules include sufficient preparation time as well as sufficient time for the formal testing.

Formal testing consists of executing the test procedures in their specified test environment, with quality assurance and possibly acquisition team witnesses. Formal testing is always preceded by a test readiness review (TRR) to ensure that the contractor is ready for the witnessed testing to begin. The acquisition team must review the contractor's readiness at the TRR and must ensure the testing is well controlled, with no hardware or software changes made "on the fly" during the formal test process, with all deviations from the test procedures redlined on the official as-run procedures, and with all encountered problems logged and documented on problem reports.

There are many stakeholders for the test planning and execution phase. Government stakeholders include the acquisition team, higher-level acquisition organizations, users and operators of the system to be acquired, specialty engineering organizations, logistics organizations, owners of interfacing systems, test and evaluation personnel, sustainment personnel, security personnel, and personnel responsible for the enterprise. There are also numerous contractor stakeholder organizations, including the contractors under contract in the EMD and PD phases and their subcontractors, and various organizations within the contractors and subcontractors, such as systems engineering, system

integration and test, specialty engineering, logistics, and sustainment organizations.

14.6.6 Transition to Operations

The transition to operations phase is usually executed during the deployment phase. It includes all of the tasks necessary to deploy the ground segment to the field and to eventually turnover the system to its owners and operators. If the ground segment has multiple installations or mobile resources that must be fielded around the world, the contractor must deliver and install each part of the ground system in its required location and then must perform testing to check out the functionality of that part of the ground system to ensure it is performing as required in its operational environment.

Preparation for transition to operations must begin long before the transition to operations phase. First, the RFP for the development of the ground segment must include statement of working tasking to support the transition to operations and to prepare for this transition. The development contractor's preparation tasks include preparation of operations manuals, support of the validation of operations manuals by the government logistics organization, preparation of training materials, and delivery of training to all operations personnel. For some ground systems, development of a training system may be part of the contract requirements. In addition, operations procedures and operational databases must be developed and validated before the turnover. The operations procedures and operational databases may be developed by the operational organization or may be on contract for the development contractors to prepare in conjunction with the operational organization.

Another area that must be complete before turnover to operations is operational testing (see above under the test planning and execution phase). Also, the on-site hardware and software maintenance personnel must be trained and hardware and software maintenance procedures must be prepared and validated (see below under the maintenance and sustainment phase).

Finally, before the transition to operations is complete, an assessment of readiness for the turnover to operations is performed.

There are many stakeholders for the transition to operations phase. Government stakeholders include the acquisition team, higher level acquisition organizations, users and operators of the system to be acquired, logistics organizations, owners of interfacing systems, test and evaluation personnel, sustainment personnel, security personnel, and personnel responsible for the enterprise. There are also numerous contractor stakeholder organizations, including the contractors under contract in the EMD and PD phases and their subcontractors, and various organizations within the contractors and subcontractors, such as systems

engineering, system integration and test, logistics, and sustainment organizations.

14.6.7 Maintenance and Sustainment

The maintenance and sustainment phase is principally executed during the Operations and Support phase. However, it must begin as soon as any part of the ground segment becomes operational, and therefore, it frequently begins during the Deployment phase. The maintenance and sustainment phase includes all tasks necessary to sustain the ground segment hardware and software in their full operational condition and to implement enhancements to the operational ground segment as requirements for such enhancements are defined and funded. Maintenance and sustainment includes refreshing the COTS hardware and software and other technology refresh, as well as fixing problems (e.g., replacing failed hardware components and fixing software bugs) and implementing small or major enhancements.

Preparation for the maintenance and sustainment phase must begin long before the maintenance and sustainment phase begins. First, the RFP for the development of the ground segment must include statement of working tasking to prepare for the turnover of the ground segment to the maintenance organization and to prepare all of the products needed to support sustainment. Maintenance manuals and procedures must be developed and validated for both hardware and software, and training materials for hardware and software maintenance must be developed and training classes delivered. In addition, the development contract must require the contractor to maintain the documentation of the ground segment requirements, architecture and design, including the requirements, architecture and design of the individual hardware and software items, to accurately reflect the as-built ground segment, so that the maintenance organization has a complete and current description of the ground segment when it begins sustainment.

Frequently the development contract includes implementation of one or more ground segment test facilities and maintenance environments and tools that are to be used when maintaining the hardware and software. Usually, maintenance is performed by the development contractor until the ground segment maintenance is turned over to the maintenance organization. Sometimes there are two maintenance organizations, one for hardware and another for software, and the turnovers for hardware and software can occur at different times.

The contractual TRD for the ground segment development should contain requirements for supportability for both hardware and software. This will cause the development contractors to build-in the ability of the ground segment to be maintained, including both its hardware and software, when they are developing the ground segment architecture and design. Specifying supportability

requirements will help ensure that the ground segment will be able to be maintained effectively and efficiently. Such requirements include both quantitative performance requirements (e.g., Mean Time to Restore Function, Mean Time to Repair) and functional requirements (e.g., inclusion of integrated diagnostics for both hardware and software).

Before the ground segment is turned over to the maintenance organization, it is recommended that the ability of the hardware and software maintainers to maintain the ground segment be demonstrated in the maintenance environment, using the documented maintenance procedures. An assessment of readiness for turnover to the maintenance organization should be performed, including the maintenance demonstration and an evaluation of the correctness and completeness of the maintenance products and procedures and the accuracy and completeness of the as-built ground segment documentation.

There are many stakeholders for the maintenance and sustainment phase. Government stakeholders include the acquisition team, higher level acquisition organizations, users and operators of the system to be acquired, logistics organizations, owners of interfacing systems, test and evaluation personnel, the maintenance organization and its maintenance personnel, security personnel, and personnel responsible for the enterprise. There are also numerous contractor stakeholder organizations, including the contractors under contract in the PD and OS phases and their subcontractors, and various organizations within the contractors and subcontractors, such as systems engineering, system integration and test, logistics, and sustainment organizations.

14.7 References

1. Adams, R. J., et al., *Software Development Standard for Mission Critical Systems*, TR-RS-2015-00012, The Aerospace Corporation, El Segundo, CA. March 17, 2014 (also known as SMC-S-012).

2. CMMI Product Team, *CMMI® for Development, Version 1.3 (CMMI-DEV, V1.3)*, Carnegie Mellon University (CMU)/Software Engineering Institute (SEI), CMU/SEI-2010-TR-033, November 2010.

3. Department of Defense (DOD), *Operation of the Defense Acquisition System*, Department of Defense Instruction DODI 5000.02, 7 January 2015.

4. Guarro, S. B., G. A. Johnson-Roth, and W. F. Tosney, *Mission Assurance Guide (MAG)*, TOR-2007(8546)-6018 Rev B, The Aerospace Corporation, El Segundo, CA. June 1, 2012.

5. Abelson, Linda, A., et al., *Integrating Software Topics into the Request for Proposal.* TOR-2011(8506)-117, The Aerospace Corporation, El Segundo, CA. July 19, 2012.

6. International Standards Organization (ISO) / International Electrotechnical Commission (IEC) / IEEE, *Systems and software engineering – System life cycle processes*, ISO/IEC/IEEE 15288-2008, 21 March 2008.

14.8 Bibliography

Institute for Electrical and Electronic Engineers (IEEE), *Draft Standard for Application of Systems Engineering on Defense Programs*, IEEE P15288.1/D4.1, September 2014.

IEEE, Draft Standard for Technical Reviews and Audits on Defense Programs, IEEE P15288.2/D5.2, September 2014.

Lutton, David, et al. *Test Requirements for Ground Systems*, TR-2013-00215, The Aerospace Corporation, El Segundo, CA. June 4, 2013.

Peresztegy, L. B. and C. E. O'Connor, *Technical Reviews and Audits of Systems, Equipment, and Computer Software*, TOR-2007(8583)-6414, Rev. 1, Vol. 1, The Aerospace Corporation, El Segundo, CA. January 30, 2009 (also known as SMC-S-021).

Peresztegy, L. B. and C. E. O'Connor, *Technical Reviews and Audits of Systems, Equipment, and Computer Software*, TOR-2007(8583)-6414, Rev. 1, Vol. 2, The Aerospace Corporation, El Segundo, CA. January 30, 2009.

Eslinger, Suellen, L. J. Holloway, and R. M. Wilkes, *Space Segment Software Readiness Assessment,* TOR-2011(8591)-20, The Aerospace Corporation, El Segundo, CA. June 3, 2011.

14.9 Acronyms

AoA	analysis of alternatives
ASP	acquisition strategy plan
CARD	cost analysis requirements document
CCA	Clinger Cohen Act
CDD	capability development document
CDR	critical design review
CDRL	contract data requirements list
CMMI®	Capability Maturity Model® Integration
CMMI®-DEV	CMMI® for development

CMU	Carnegie Mellon University
CONOPS	concept of operations
COTS	commercial off-the-shelf
CPD	capability production document
CSCI	computer software configuration item
CTE	critical technology element
DOD	Department of Defense
DODI	DOD instruction
EMD	engineering and manufacturing development
FD	full deployment
FDD	full deployment decision
FFRDC	Federally Funded Research and Development Center
FOC	full operational capability
GRA	government reference architecture
I	initial preparation
ICD	initial capabilities document
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IOC	initial operational capability
IP	intellectual property
ISP	information support plan
ISO	International Standards Organization
LCSP	lifecycle support plan
MAG	Mission Assurance Guide
MSA	materiel solution analysis
OS	operations and support
OT&E	operational test and evaluation
PD	production and deployment
PDR	preliminary design review
PESHE	programmatic environment, safety, and occupational health evaluation
PPP	program protection plan
RFP	request for proposal
RS	recommended standard
S	standard
SAR	software architecture readiness
SDR	system design review
SE&I	systems engineering and integration
SEI	software engineering institute
SEP	systems engineering plan
SETA	system engineering and technical assistance
SFR	system functional review
SMC	Space and Missile Systems Center
SRD	system requirements document
SRR	system requirements review

SS	system specification
SSDD	system/segment design document
STAR	system threat assessment report
SWAMP	software acquisition management plan
TDS	technology development strategy
TEMP	test and evaluation master plan
TES	test and evaluation strategy
TMRR	technology maturation and risk reduction
TOR	Aerospace technical operating report
TR	Aerospace technical report
TRA	technology readiness assessment
TRD	technical requirements document
TRR	test readiness review
U	update

Chapter 15
Concept Development, RFP, and Source Selection

David J. Naiditch
Software Acquisition and Modeling Department
Software Engineering Subdivision
Geraldine A. Chaudhri
Software Systems Assurance Department
Computer Applications and Assurance Subdivision

15.1 Introduction/Background

We describe the best practices, key lessons learned, and a set of resources to support work to be completed in the initial phases of Ground Segment acquisitions. There are three main activities that must take place in these initial phases: (1) develop the concept, (2) prepare and submit the request for proposal (RFP), and (3) conduct the source selection. In addition, after the concept is developed, there are two phases that a program enters that help in the decision making process: the materiel solution phase and the technology development phase.

15.2 Definitions

Contract The legally binding agreement between the program and the contractor.

Contractor A person or organization that enters into a contract with the program for the supply of a product. The term "contractor" is intended to provide a neutral and broad definition of acquisition that includes those delivering products or performing services as well as those contracted (such as a prime contractor) to develop and deliver products.

CDRL (Contract Data Requirements List) The itemization of the development products to be delivered by the contractor to the program as part of the contract.

CMMI® Capability Maturity Model Integration A process improvement training and appraisal program and service administered and marketed by The Software Engineering Institute of Carnegie Mellon University and required by many DOD and U.S. Government contracts, especially for software development.

Offeror A person or organization that responds to a request for products or services, but is not yet on contract for those services. See also "contractor."

Prime Contractor The contractor that has a contract directly with the government. The prime contractor may contract with "subcontractors" to perform part of the technical effort of the contract. This document refers to the prime contractor and the subcontractors as "contractors".

15.3 Summary of Tasks

Figure 15-1 illustrates the activities and the associated tasks that are performed during this initial phase of the acquisition cycle.

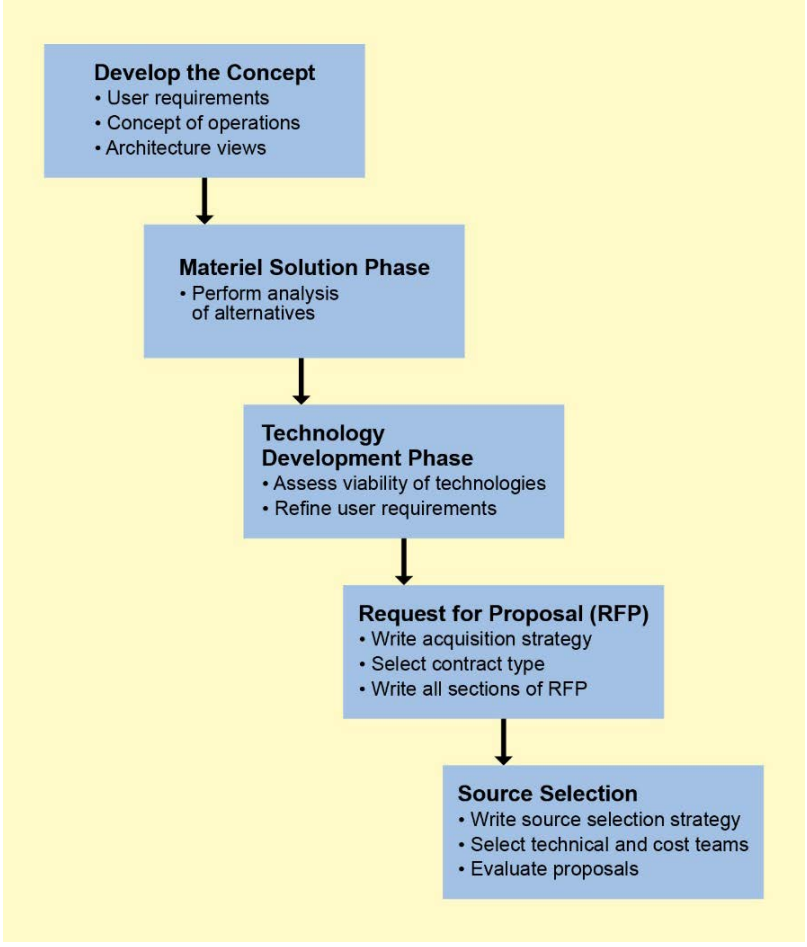


Figure 15-1. Summary of tasks in the initial phase of the acquisition cycle.

15.3.1 Develop the Concept

Development of the concept for a ground segment acquisition begins with the development of a set of user requirements, which are usually documented in a Technical Requirements or a Concept of Operations document. These documents may also be supplemented with DODAF views. The Department of Defense Architecture Framework (DODAF) provides visualization infrastructure for specific stakeholder concerns organized by various views [1].

After the user requirements, concept of operations, and architecture views are complete, the materiel solution and the technology development phases begin.

15.3.1.1 The Materiel Solution Phase

The purpose of this phase is to assess potential materiel solutions by performing an Analysis of Alternatives (AoA). The AoA is required by DoD 5000.02 for all acquisition category (ACAT) I programs and may be ordered for ACAT II and III programs [2]. Air Force instruction (AFI) 10-601, Capabilities-Based Requirements Development, specifies the AoA process and responsibilities [3]. The AoA is an objective and detailed comparison of various alternatives for meeting technological requirements. The AoA plays a critical role in determining whether a system should be procured, and if so, identifying the most cost-effective, beneficial, and least risky alternatives for procuring the needed capabilities. The AoA is examined by senior Air Force leaders to determine a program's capability and affordability [4]. In addition, the AoA guides the source selection criteria.

At the close of the AoA, the program office takes ownership of the approach and conducts additional engineering analysis to support the development of the technical development strategy (TDS), the test and evaluation strategy (TES), and the systems engineering plan (SEP). It is critical that the program office's systems engineering analysis builds upon the AoA results and provides the program manager with the technical basis for executing the technology development phase, including risk-reduction efforts. In particular, during the material solution phase, the systems engineering team performs the following activities: [3]

- Confirm the concept of operations (CONOPS) and develop mission and functional threads with users
- Develop initial view of system requirements and system design concepts
- Identify critical technology elements (CTEs)
- Determine external interfaces and interoperability
- Identify critical program information (CPI)

15.3.1.2 The Technology Development Phase

The technology development phase incorporates the objectives of buying down technical risk and developing a sufficient understanding of possible solutions to make a sound decision about initiating a formal acquisition program [5]. It is a continuous technology discovery and development process reflecting close collaboration between the science and technology community, the user, and the system developer. It is an iterative process designed to assess the viability of technologies while simultaneously refining user requirements.

Acquisition and program risks need to be identified before the RFP is developed, because these risks play a key role in determining the RFP content. For example, the RFP should require prototyping of identified high-risk technologies, such as complex new algorithms and mission planning concepts, difficult timing requirements, and novel mobile ground stations.

Acquisition and program risks are identified when performing market research, developing the acquisition strategy, writing the source selection plan, and identifying critical hardware and software technology elements. Given the importance of software to the success of today's ground systems, it is essential for software-cognizant personnel to participate in risk analysis and provide early support to the development of the acquisition strategy and source selection plan.

Products produced during this phase include the acquisition strategy, the cost acquisition requirements description, and the independent cost estimate.

15.3.2 Request for Proposal (RFP)

The RFP is used by the government to solicit proposals from industry. An RFP follows the Uniform Contract Format (UCF) as specified in Federal Acquisition Regulation (FAR) 15.204-1, Table 15-1 [5]. As shown in Table 15-1, the UCF has four parts and 13 sections. For this handbook, the sections of special interest are sections H, J, L, and M.

Table 15-1. Uniform Contract Format

Section	Title
	Part I—The Schedule
A	Solicitation/Contract Form
B	Supplies or Services and Prices/Costs
C	Description/Specifications/Statement of Work
D	Packaging and Marking
E	Inspection and Acceptance
F	Deliveries or Performance
G	Contract Administration Data

Section	Title
H	Special Contract Requirements
	Part II—Contract Clauses
I	Contract Clauses
	Part III—List of Documents, Exhibits, and Other Attachments
J	List of Attachments
	Part IV—Representations and Instructions
K	Representations, Certifications, and Other Statements of Offerors or Respondents
L	Instructions, Conditions, and Notices to Offerors or Respondents
M	Evaluation Factors for Award

Section A contains administrative information such as the RFP number, proposal due date, and government points of contact. Section B contains a brief description of each contract deliverable covered by a contract line item number (CLIN). There should be a CLIN for every item or major category of service supplied by the contractor to the government. Section C contains a more detailed description of the items contained in Section B. It describes what tasks are to be performed, but not how these tasks are to be performed. In most RFPs, this section references appendices or exhibits for the actual content. Section D contains specific information on requirements for packaging and marking of delivered items, such as preservation, protection, and bar coding. Section E contains information on how the government will inspect delivered items, and contains conditions for acceptance of delivered items under the contract, such as place of inspection, quality assurance, reliability requirements, whether a receiving report is required, who will inspect the items, and acceptance criteria. Section F specifies the requirements for time, place, and method of delivery or performance for items to be delivered under the contract. Section G contains accounting and appropriations data and required contract administration information and instructions.

Section H contains special contractual requirements not included in other sections of the RFP. These special requirements may include requirements for the Electronic Data Interchange (EDI) Network that provides the government with electronic access to contractor information such as earned value management (EVM), the work breakdown structure (WBS), cost performance report (CPR), integrated master schedule (IMS), and technical performance measurements (TPMs). Section H may also include requirements for CMMI® appraisals (see “Capability Evaluations,” section 19.3.4) [6], architecture evaluations that provides the right for the government to perform independent architecture evaluations during the contract, and The Aerospace Corporation (Aerospace) enabling clauses. Enabling clauses ensure that Aerospace will have access to subcontractor technical data needed to perform systems engineering and integration for the program office. Section H also specifies requirements for

data rights in technical data and software (see section 19.3.2.1), information assurance, and contract requirements.

Section I contains all clauses required by law or regulation. These clauses are commonly referred to as "boilerplate" clauses and are normally inserted into most contracts by the procuring contracting officer (PCO). Section J provides a list of attachments and exhibits that are a material part of the contract. Attachments may include technical specifications, statement of work (SOW) or performance work statement (PWS), process documents, and contract data requirements list (CDRL) items. Section K contains representations, certifications, and other information required from each offeror.

Section L guides offerors in preparing their proposals by including instructions to offerors (ITO) and solicitation provisions. This section contains the requirements for what is to be provided in a proposal such as the executive overview, integrated master plan (IMP) that is used to track the program's status, and the integrated master schedule (IMS) that includes the schedule for the contractor and all subcontractors. Section L also lists required attachments such as software development plan (SDP), software architecture description (SAD), past performance questionnaire, systems engineering management plan (SEMP), basis of estimates (BOEs), contractor statement of work (CSOW), and rights in technical data and software.

Section M matches the structure of L. It explains the basis for evaluating each offeror's proposal. It provides the evaluation criteria and prioritizes their importance, as well as the evaluation methodology, so that an independent and objective evaluation can be made of each offeror's proposal

The following sections specify important topics of the RFP: data rights, Awards and Incentives, Cost Volume, Commercial Off the Shelf (COTS) Products, Measurement, and CDRL.

15.3.2.1 Data Rights

Ground systems make extensive use of commercial components that are generally acquired from either open source or commercial providers. As a result, ground systems rely heavily on commercial item data rights. Data rights, or more technically, "Government rights in technical data and software," include the rights to use, modify, release, produce, display, or disclose technical data or software. Technical data includes technical specifications, user manuals, and maintenance manuals about hardware and software components of the acquired system. Commercial software includes components such as virtual machines, operating systems, networking software, system monitoring, graphical user interfaces, and databases. Ensuring that the government acquires adequate rights in technical data and software delivered under the contract is crucial for both

acquisition oversight and subsequent sustainment. The rights in software and technical data that a program office receives will have a significant impact not only during the development phase but also (and perhaps especially) during the operations and maintenance phase. Specifically, the RFP must include requirements to ensure that government, Federally Funded Research and Development Center (FFRDC), System Engineering and Technical Assistance (SETA), and systems engineering and integration (SE&I) personnel will have timely access to all contractor data needed to independently manage and oversee the ground system during contract execution. Without this data, the Government's ability to identify and mitigate program risks and to fix problems would be severely compromised, possibly resulting in negative impacts to cost, schedule, performance, and quality. Furthermore, the government must have access to technical data needed to operate and maintain the ground systems.

The Defense Acquisition Regulations System (DFARS) 227.7207 [7] states, "Commercial computer software or commercial computer documentation shall be acquired under the licenses customarily provided to the public unless such licenses are inconsistent with federal procurement law or do not otherwise satisfy user needs." Licenses apply to privately copyrighted material (including both software and technical data). Obtaining a license is not the same as ownership of that material (even transferable or unlimited licenses) because in a license, the private owner retains the copyright. DFARS also states that offerors and contractors shall not be required to furnish information or rights that they do not offer to the public. Furthermore, for technical data, the government may not use or release such data outside of the government unless such use, release, or disclosure (1) is necessary for emergency repair/overhaul of the commercial items procured, (2) is form/fit/function data, (3) is a correction, (4) is a change to the technical data furnished to the contractor by the government, or (5) permission is obtained by the government from the licensor (through purchase or negotiation).

Although ground systems make extensive use of commercial components, non-commercial components are also employed. Unfortunately, data rights for non-commercial components are very complex and data rights experts need to be consulted early in the acquisition. The following discussion just provides an overview of the issues that arise.

For non-commercial technical data and software, contracts incorporate the following standard levels of data rights [7].

"Unlimited rights" are very broad rights tantamount to complete ownership.

"Government purpose rights" (defined in the DFARS but not the FAR) allow for unrestricted use of the information within the Government and for any activity

in which the U.S. Government is a party (in particular, this refers to foreign military sales).

"Limited rights" (for technical data) permit the use only within the Government except for "form, fit, and function" (in essence, interface) data and emergency repair and overhaul information to support (not development) contractors.

"Restricted rights" (for non-commercial software) permit the Government to use, release, and disclose such software outside the Government to contractors or subcontractors (1) "performing service contracts to diagnose and correct deficiencies in a computer program, to modify computer software to enable a computer program to be combined with, adapted to, or merged with other computer programs or when necessary to respond to urgent tactical situations" provided that certain procedural conditions have been satisfied, or (2) "performing emergency repairs or overhaul" provided that certain procedural conditions have been satisfied.

"Specially negotiated rights" are other situations not fitting into the above categories.

DFARS 227.7103-5 and 227.7203-5 define data rights by the source of funds used to develop the software. Table 15-2 shows the standard terms.

Table 15-2. Source of Funds and Government Rights

Source of Funds	Rights
Government	Unlimited
Mixed	Government purpose for a negotiated period, after which the Government receives unlimited rights
Private	Limited (item/process) or restricted (computer software)
Other	Special purpose

In general, the RFP cannot mandate the delivery of technical data or computer software with unlimited or government purpose rights for products developed with non-government funding, but the RFP can be structured to encourage the offerors' granting of technical data and computer software rights by means of the instructions for preparation of the proposal and the evaluation criteria.

One of the goals of the software data management strategy is to specifically the minimum necessary rights in technical data and software for each anticipated software-related item or class of items in the program. In its first iteration (when developed as part of the acquisition strategy), not enough may be known about

the program to fully reach this goal. Thus, the strategy should be revisited when the RFP is being written. At least two areas should be reviewed: (1) systems engineering, integration, and system-level testing, and (2) software architecture, design, and implementation.

Section B of the RFP contains CLIN for the purchase of data rights. Section H includes language giving the government the option to purchase data rights. Section I incorporates data rights clauses into the contract. Section J defines an attachment for the listing of data rights. Section K requires the offeror to specifically identify all restrictions on data rights that are not unlimited. Section L requires offeror to enumerate the data rights on software and technical data and cost of these rights and identify gaps between what the offeror is offering if less than what the government is requiring. Section M includes an evaluation criterion that discriminates among offerors based on proposed data rights.

15.3.2.2 Contract Type, Awards, and Incentives

There is a wide selection of contract types available to the government and contractors when an acquisition is planned. The contract types are grouped into two broad categories: fixed-price contracts and cost-reimbursement contracts. The specific contract types range from firm-fixed-price, in which the contractor has full responsibility for the performance costs and resulting profit (or loss), to cost-plus-fixed-fee, in which the contractor has minimal responsibility for the performance costs and the negotiated fee (profit) is fixed. In between are the various incentive contracts, in which the contractor's responsibility for the performance costs and the profit or fee incentives offered are tailored to the uncertainties involved in contract performance.

It is necessary to use the type of contract that establishes appropriate responsibility, accountability, risk, and reward on the contractor to motivate good performance, yet relieves the contractor of excessive risks over which it has no control.

For ground segment acquisition, certain contract types may be more suitable depending on the development phase. For example, a new development with poorly defined requirements is not well suited for fixed price since there is excessive contractor risk. Sustainment of a well-documented system may be more suitable to fixed price.

In concert with the basic contract type, the following other incentives need to be carefully considered: award fee incentives, cost/financial incentives, performance incentives, schedule/delivery incentives, and multiple incentives.

The goal of contract incentives is to motivate optimal contractor performance in areas deemed critical to an acquisition program's success: cost, schedule,

performance, and quality. The Federal Acquisition Regulation (FAR) Subpart 16.4 prescribes policies and procedures for incentive contracts. It discusses the major types of contract incentives and gives guidance on using multiple incentives in the same contract [5].

Award Fee Incentives: the contractor's ability to earn a bonus or "fee" based on exceptional performance. The award fee is earned by the contractor based on the government's assessment of the contractor's performance in relation to specific criteria documented in the award fee plan.

Cost/Financial Incentives: A cost incentive relates profit or fee directly to results achieved by the contractor. These incentives are normally based on a shared formula between the government and the contractor [i.e., fixed-price incentive (FPI) or cost plus incentive fee (CPIF) contracts] or the payment of a fee from an award fee pool. To be effective the incentives must be quantitative, clearly related to the desired outcome, and within a reasonable range. The rewards must be commensurate with the risks the contractor assumes. Cost incentives for ground segment acquisitions might entail awarding money when milestones are met by the due date or when cost is contained within a specific threshold.

Performance Incentives: Performance incentives are designed to relate profit to the contractor's achieved results based on specified targets. Performance incentives (either positive or negative) should be used when they will induce better quality performance. A performance incentive should be applied selectively to motivate efforts that may not otherwise be adequately emphasized and to discourage inefficiency. Incentives should apply to the most important aspects of the work, rather than to each individual task. Incentivizing too many requirements dilutes the monetary importance of each requirement and creates an administrative burden for the government. Performance incentives for ground segment acquisitions would include rewards for producing software and documentation with a low defect density.

Schedule/Delivery Incentives: Schedule incentives focus on getting a contractor to meet or exceed minimum delivery requirements. They can be defined in terms of early delivery, attaining or exceeding milestones, or meeting rapid-response or urgent requirements.

Multiple Incentives: Any contract that contains more than one incentive is a multiple incentive contract. If multiple incentives are used, the amounts allocated to each incentive and fee area must be sufficient to adequately motivate and reward a contractor to excel in each. A balance is needed so no incentive is too insignificant that it offers little reward for the contractor or so large that it overshadows all other incentives. All multiple-incentive contracts must preclude rewarding a contractor for superior technical performance or

delivery results when the cost of those results outweighs their value to the Government (See FAR Part 16.402-4) [5].

15.3.2.3 Cost Volume

Section L of the RFP should require the offeror to provide a detailed cost estimate in the cost volume. The government needs to independently evaluate the cost volume for reasonableness and realism with the ground system architecture proposed in the technical volume. Furthermore, the government needs to verify that the cost of data rights is consistent with the offeror's data rights proposal. Data rights include rights to technical data such as specifications, maintenance manuals and user manuals, as well as rights to the software, including operational software and software embedded in support tools and test equipment. The evaluation of technical risk in an offeror's ground system approach is linked to the cost realism analysis of the proposed ground system cost volume.

15.3.2.4 Commercial Off the Shelf (COTS) Products

Because so many COTS products are used in ground systems, sections L and M of the RFP should require offerors to describe how they plan to mitigate the risks associated with using COTS products. Considerations include:

- Evaluating the quality and adequacy of COTS products before purchasing them or integrating them into the system.
- Ensuring the use of standard interfaces to COTS products to minimize dependence on products of a particular vendor. If a non-standard interface is proposed, consider building an isolation layer, with standard interfaces on the application side.
- Evaluating how an offeror will respond if a software vendor goes out of business or stops supporting the COTS product.
- Identifying and justifying the customization or modification of any COTS product. (This is strongly discouraged because the software would need to be modified repeatedly each time a new version is released.)
- Identifying threshold and objective requirements to convey to the offeror which requirements can be relaxed to achieve a commercial-based solution.
- Structuring the cost volume to capture possible increased costs for licenses or support services.
- Planning to test each COTS software upgrade to ensure the new features work as expected in the target environment and that no new bugs have been inadvertently introduced.

- Maintaining currency with COTS software upgrades during the development and sustainment phase with minimal disruption.
- Procuring needed technical information that is not generally provided by the commercial software vendor.
- Verifying that sufficient data rights are granted to use COTS as required.

Many of the risks described for COTS also apply to the other non-developmental software items too.

15.3.2.5 Open Source Software

Open source software is freely available as source code. Typically, a software license permits users to modify and distribute the software. Some open source software is in the public domain and developed in a public, collaborative manner. If open source software is used, the license must be carefully examined to determine whether the terms are acceptable. In particular, care must be taken if the license requires modified open source software to be rereleased for public use.

15.3.2.6 Measurement

Metrics are a critical management tool for the successful acquisition of large systems. Metrics offer management visibility into both the acquisition process and the development process by providing insight about the progress, quality, and expected completion of a development effort. A metrics collection activity should be defined for both the program office and the contractor. The metrics collection requirements for the contractor is specified in the software measurement report data item description (see the section on CDRL for more information). For the program office, the metrics collection plan would be specified in the program management plan or software acquisition management plan.

Three tasks must be performed to define an effective measurement program: Determine the scope of the measurement program, determine the collection and reporting level of the measurement program, and specify the required measurement data.

The first task is to determine the scope of the measurement program, that is, will the measurement program be software only, systems only, combined software and systems, or separate software and systems. How the government plans to organize determines this scope. If a strong systems engineering discipline exists, a separate systems-only measurement plan and measurement report may be

needed. In addition, a systems engineering measurement specialist may be required.

The second task is to determine the collection and reporting level of the measurement program. At too high a level, the measurement data does not provide enough insight into the program risk; at too low a level, the measurement data will be costly to collect and report. Critical for this decision is an understanding of the system and software architecture and the system and software integration build-up strategy. Often the collection and reporting levels are defined in the proposal by the bidding contractors, which can cause the measurement reporting to be inadequate and inconsistent. The authors recommend that the government define the collection and reporting levels in the RFP.

The third task is to specify the required measurement data. The authors recommend that the government select the measurements for the contractor to collect and report for each program, rather than leave this decision to each offeror. This ensures that the measurement programs across offerors are comparable and that the collected and reported data is sufficient to manage the program.

15.3.2.7 Contract Data Requirements List (CDRL)

CDRL items are a list of required delivered documentation and data. The government uses CDRL items to specify to the contractor what data is required and when it is to be delivered. The government has no rights to contractor data unless that data is ordered through the CDRL and the contractor formally delivers that data to the government. If the government only needs temporary access to the contractor's data, for example to perform analysis or to determine status, then access to data in the digital access list (DAL) is sufficient.

The government should specify the minimum required CDRL items in the RFP. If government does not require CDRL items in RFP, the contractor will charge more to add CDRL items after contract award. Offerors, of course, are free to choose to expand the CDRL items in their proposal. The format and content of CDRL items are defined in Data Item Descriptions (DIDs). DIDs often need tailoring to meet the specific needs of the ground system. Each CDRL needs to be justified by listing the reasons the CDRL is important as well as enumerating the consequences of not having the CDRL. CDRLs may need to be justified multiple times during the RFP development. "*Recommended Software –Related Contract Deliverables for National Security Space System Programs*," TOR-2006(8506)-5738 provides expert assistance in developing and justifying the list of software CDRLS [8].

15.3.3 Source Selection

After the government releases an RFP, offerors respond by submitting their proposals. Source selection is the formal process of determining which offeror(s) is awarded the contract. The source selection evaluation board (SSEB) evaluates the offeror's proposals. The SSEB typically consists of an SSEB chair, a technical factor chief for the technical team, a factor chief who leads the past performance team (PPT), and a factor chief and team for cost/price. The source selection authority (SSA) appoints the chair for the SSEB and the past performance team. The SSEB chairperson selects the members of the technical and the cost/price team as well as the factor chiefs. The members of the SSEB are identified in the source selection plan.

15.3.3.1 Technical Team

The technical team is of central importance because this team is responsible for two ratings: the technical rating and the technical risk rating. The technical rating evaluates how well the offerors meet the RFP requirements. The technical risk rating evaluates the risk of the offerors' approach to meeting these requirements. These ratings are made for each of the technical subfactors listed in the RFP. Because ground systems are software–intensive systems, it is critical that software experts are adequately represented on the technical teams.

To ensure consistent evaluations across all proposals, the technical team is instructed to evaluate the proposals one at a time against section M criteria, and not to compare proposals to one another. Furthermore, teams only evaluate proposals based on what the RFP asks for, not on what they think should have been required. Teams are also instructed to only consider what they read in the proposal and not bring in information from other sources.

Each member of the technical team needs to carefully read sections L and M of the RFP, read the offeror's executive summary (assuming it was required), review the pertinent sections of the proposal, and evaluate their assigned subfactors against the section M evaluation criteria. If performing a full trade-off source selection, the Section M criteria is assessed as exceeding requirement, meeting requirement, not clearly meeting requirement, or failing to meet requirement. Each strength identified in the technical rating needs to be evaluated for an associated risk to schedule, cost, or performance.

One of two distinct methodologies can be used to evaluate the technical approach and related risk. Methodology 1 includes risk associated with the technical approach in a single rating. Methodology 2 provides separate technical and risk ratings. Tables 3–5 illustrate these two methodologies. The technical team chief (or subfactor chief) combines inputs from all the technical evaluators and advisors prior to determining ratings. Dissenting opinions are presented to

the SSA along with the majority opinion. Proposals with an unacceptable rating are not awardable.

Table 15-3. Methodology 1 - Combined Technical/Risk Ratings

Color	Rating	Description
Blue	Outstanding	Proposal meets requirements and indicates an exceptional approach and understanding of the requirements. Strengths far outweigh any weaknesses. Risk of unsuccessful performance is very low.
Purple	Good	Proposal meets requirements and indicates a thorough approach and understanding of the requirements. Proposal contains strengths which outweigh any weaknesses. Risk of unsuccessful performance is low.
Green	Acceptable	Proposal meets requirements and indicates an adequate approach and understanding of the requirements. Strengths and weaknesses are offsetting or will have little or no impact on contract performance. Risk of unsuccessful performance is no worse than moderate.
Yellow	Marginal	Proposal does not clearly meet requirements and has not demonstrated an adequate approach and understanding of the requirements. The proposal has one or more weaknesses which are not offset by strengths. Risk of unsuccessful performance is high.
Red	Unacceptable	Proposal does not meet requirements and contains one or more deficiencies. Proposal is unawardable.

Table 15-4. Methodology 2 - Technical Ratings Separate from Risk Ratings

Color	Rating	Description
Blue	Outstanding	Proposal meets requirements and indicates an exceptional approach and understanding of the requirements. The proposal contains multiple strengths and no deficiencies.
Purple	Good	Proposal meets requirements and indicates a thorough approach and understanding of the requirements. Proposal contains at least one strength and no deficiencies.
Green	Acceptable	Proposal meets requirements and indicates an adequate approach and understanding of the requirements. Proposal has no strengths or deficiencies.
Yellow	Marginal	Proposal does not clearly meet requirements and has not demonstrated an adequate approach and understanding of the requirements.
Red	Unacceptable	Proposal does not meet requirements and contains one or more deficiencies and is unawardable.

Table 15-4. Risk Ratings Separate from Technical Ratings

Rating	Description
Low	Has little potential to cause disruption of schedule, increased cost or degradation of performance. Normal contractor effort and normal government monitoring will likely be able to overcome any difficulties.
Moderate	Can potentially cause disruption of schedule, increased cost, or degradation of performance. Special contractor emphasis and close government monitoring will likely be able to overcome difficulties.
High	Is likely to cause significant disruption of schedule, increased cost, or degradation of performance. Is unlikely to overcome any difficulties, even with special contractor emphasis and close government monitoring.

Technical risk evaluations examine any weakness with an offeror's approach that increases the risk of unsuccessful contract performance. The evaluations can manifest in having the potential to cause disruption of schedule, increased cost, or degradation of performance. The technical team assesses the criteria by identifying weaknesses and significant weaknesses as defined in FAR 15.001 [6]. Risks might arise from proposed untested new technologies, unproven processes, or novel uses of technologies. For each weakness, an evaluator creates an evaluation notice (EN) and evaluates the response to determine whether the offeror has provided appropriate risk mitigation.

After the SSEB presents their evaluations, discussions with the offerors in the competitive range determined by the SSA will occur to resolve ENs and to share ratings with the offerors. Offerors may be permitted to update their proposals to deal with the identified deficiencies, requirements not clearly meet, or weaknesses. A request for the final proposal revisions may be issued at the conclusion of all discussions with the offerors. The SSEB then evaluates the final proposals. The SSA is presented the final ratings that include the technical and risk rating for each subfactor, past performance rating, and cost/price analysis. After being presented to the SSA, all evaluative material and supporting material enters the official record and must be maintained. Debriefings are conducted separately for each offeror that requests one. At the debriefing, the offeror's strengths, weaknesses, and deficiencies are provided among other information.

15.3.3.2 Cost/Price Team

In addition to the technical team, there is the cost/price team. The cost/price team evaluates the reasonableness and realism of contractor's proposed cost/price for work to be performed. This cost evaluation is based on factors

such as cost models, market information, and recent program history. An independent cost estimate is prepared by the program office prior to the release of the RFP. The program office documents this cost estimate in the cost analysis requirements description (CARD) and uses it to evaluate the accuracy of the contractor's cost estimate.

15.3.4 Capability Evaluation

SMC policy requires a software capability evaluation be performed on all offerors as part of a source selection for software-intensive systems, such as new development or sustainment of the ground segment. The Engineering Institute's Standard CMMI® Appraisal Method for Process Improvement (SCAMPI) [6] is the preferred method. A software capability evaluation aims to identify and mitigate software risks that can result in cost increases, schedule slips, and unacceptably low-quality software products. Note that although CMMI originated in software engineering, it has been expanded and generalized to include the development of hardware and the delivery of services, as well as the acquisition of products and services. Therefore, although SMC policy only requires a software capability evaluation, it can be expanded to include a hardware capability evaluation too.

The purpose of performing a capability evaluation is to determine if the offeror has defined and disciplined product development processes that produce repeatable results, are consistent with the Capability Maturity Model Integration (CMMI®), and are used by the entire team throughout the period of performance. A defined process includes readiness (entrance) criteria, inputs, standards, procedures for performing the work, verification methods used to ensure process performance and product quality (e.g., peer reviews), outputs, process and product measures, and completion (exit) criteria. A defined process identifies the roles and responsibilities of the participants and stakeholders. A defined process is planned, employs skilled people having adequate resources, involves relevant stakeholders, is monitored and controlled, and is evaluated for adherence to its process description.

The authors do not recommend requiring a specific CMMI maturity level rating, because such ratings are organizational, not for a program. The authors recommend that Section H of the RFP requires that the offeror have a defined and disciplined product development process that produces repeatable results, is consistent with the CMMI, and is used by the entire team throughout the period of performance. A defined process includes the attributes defined above. The process must be integrated across all team members in a manner described by the process architecture.

SCAMPI appraisals can be performed in three levels of formality: SCAMPI A, B, or C. Both SCAMPI A and B require interviews as part of the process, and

therefore cannot be used during source selection. However, a SCAMPI C appraisal of the development processes can be conducted that is solely based on a review of artifacts submitted with the proposal. In other words, a SCAMPI C can be performed during source selection because it does not require interviews to support the appraisal. The RFP instructs the contractors to provide their documented program processes, along with evidence of use (artifacts), and their process architecture. The process architecture describes the ordering, interfaces, interdependencies, and other relationships among the process elements and between process elements and external processes (for example, government or subcontractor processes). An experienced SCAMPI lead is required on the source selection team to perform the SCAMPI appraisal.

If a software capability evaluation is to be performed during source selection, Section L of the RFP states that the government will perform a SCAMPI appraisal on the offeror as part of source selection. The scope of the appraisal is described, including what process areas will be covered. As previously mentioned, the scope of the appraisal may include topics outside of software, including systems engineering, program management, and supplier sourcing. Section L instructs the offeror to provide their documented program process, their process architecture, and process artifacts. Section L also provides specific instructions to the offeror about the process descriptions, the artifacts, and the process architecture. The offeror team may be required to provide artifacts from each of the team members. If the offeror's defined software development processes have never been executed, the RFP may require artifacts from previous programs that are similar in nature. Section M of the RFP describes how the offeror's process maturity will be evaluated for contract award.

15.3.4.1 Past Performance

The PPT is a member of the source selection evaluation board tasked with evaluating each offeror's past performance through an analysis of the offeror's recent, relevant performance to determine a performance confidence assessment rating as defined in the DOD Source Selection Procedures (Table 5). A questionnaire is often used to obtain information on past performance from each of the Offerors. Questions should relate to technical and price/cost factors. ENs are prepared for negative past performance and Offerors are allowed to respond to these ENs. The PPT evaluates the Offerors' responses and determines whether or not the purported negative past performance should be considered a risk to the current contract.

15.4 Key Lessons Learned

These initial phases of the acquisition cycle are perhaps the most critical because they define the product/s to be acquired. The development of the concept and the documentation of the technical requirements begin the whole cycle. Then the

RFP and source selection phases set the ground rules for success and establish a contractual commitment to move forward with the acquisition. One of the most important and key tasks is the development of the technical requirements. Without a good set of requirements, there is no foundation upon which to build a product. It is akin to building a house. If the foundation is weak, the house will crumble. It is therefore extremely important to ensure that the technical requirements are complete, well understood, and agreed upon by all stakeholders before moving forward with an acquisition. This leads to a smoother process in the development cycle and avoids requirements creep and their resulting cost and schedule increases.

Another key lesson learned is to determine realistic and independent estimates for the work to be performed by the contractor. In the program's early acquisition life cycle phases, cost and schedule estimates should be based upon historical data collected in similar past programs. With a realistic cost and schedule estimate as a reference, a successful source selection can be accomplished, and consequently a successful acquisition.

15.5 References

1. Chief Information Officer, Department of Defense. The DODAF Architecture Framework version 2.02, Change 1. January 2015. http://dodcio.defense.gov/TodayinCIO/DoDArchitectureFramework.aspx

2. Analysis of Alternatives (AoA) Handbook: A Practical Guide to Analyses of Alternatives. July 2008. Office of Aerospace Studies, Air Force Materiel Command (AFMC) OAS/A9].

3. AFI 10-601, Operational Capabilities Requirements Development. November 6, 2013.

4. DODI 5000.02, Operation of the Defense Acquisition System. Department of Defense Instruction dated January 7, 2015.

5. FAR – Federal Acquisition Regulation, Washington, D. C., DOD. January 20, 2015. See http://farsite.hill.af.mil/vffara.htm

6. Haynes, Will, Gene Miluk, Lisa Ming, Margaret Glover, and members of the SCAMPI Band C Project. Software Engineering Institute, Handbook for Conducting Standard CMMI Appraisal Method for Process Improvement (SCAMPI) B and C Appraisals, Version 1.1. Handbook. CM4/SEI-2005-HB-005. December 2005.

7. DFARS - Defense Federal Acquisition Regulation Supplement, Acquisition Regulations, Washington, D. C., DOD, 1998. See htthp://farsite.hill.af.mil, accessed March 2012.

8. Owens, Karen L. and Joanne M. Tagami. *Recommended Software –Related Contract Deliverables for National Security Space System Programs*. TOR-2006(8506)-5738, The Aerospace Corporation, El Segundo, CA. February 14, 2008.

15.6 Biliography

DFARS 215.3, Director, Defense Procurement and Acquisition Policy memorandum dated March 4, 2011. DOD Source Selection Procedures. Revised January 15, 2015.

Integrating Software Topics into the Request for Proposal. TOR-2011(8506)-117, The Aerospace Corporation, El Segundo, CA. July 19, 2011.

Dahmann, Judith, and Kelley, Mike, White paper, "*Systems Engineering during the Materiel Solution Analysis and Technology Development Phases,"* Office of the Director, Defense Research and Engineering, September 2009.

15.7 Acronyms

ACAT	acquisition category
AFI	Air Force instruction
AoA	analysis of alternatives
BOE	basis of estimates
CARD	cost analysis requirements description
CDRL	contract data requirements list
CDRL	contract data requirements list
CLIN	contract line item number
CMMI	Capability Maturity Model Integration
CONOPS	concepts of operations
COTS	commercial off-the-shelf
CPI	critical program information
CPIF	cost plus incentive fee
CPR	cost performance report
CSOW	contractor statement of work
CTE	critical technology element
DAL	digital asset management
DFARS	Defense Acquisition Regulations System
DID	data item description
DODAF	Department of Defense Architecture Framework
EDI	Electronic Data Interchange

EN	evaluation notice
EVM	earned value management
FAR	Federal Acquisition Regulation
FFRDC	Federally Funded Research and Development Center
FPI	fixed-price incentive
IMP	integrated master plan
IMS	integrated master schedule
IMS	integrated master schedule
ITO	instructions to offerors
PCO	procurement contracting officer
PPT	past performance team
PWS	performance work statement
RFP	request for proposal
SAD	software architecture description
SCAMP	Standard CMMI® Appraisal Method for Process Improvement
SDP	software development plan
SE&I	Systems Engineering and Integration
SEMP	systems engineering management plan
SEP	systems engineering plan
SETA	system engineering and technical assistance
SOW	statement of work
SSA	source selection authority
SSEB	source selection evaluaton board
TDS	technical development strategy
TES	test and evaluation strategy
TPM	technical performance measurements
UCF	uniform contract format
WBS	work breakdown structure

Chapter 16
Requirements Engineering

Suellen Eslinger
Software Engineering Subdivision
Computers and Software Division

16.1 Introduction/Background

Ground segment requirements engineering constitutes one of the major activities of ground segment systems engineering. A requirement is a condition or capability that must be met or possessed by a ground segment, subsystem, component, or hardware or software item to satisfy an agreement, contract, standard, specification, or other formally imposed documents (Ref 1. adapted from clause 3.2506 of [ISO/IEC/IEEE 2010]). Requirements engineering, is a systems engineering discipline that consists of requirements development and requirements management. Ground segment requirements development is the process of defining and documenting ground segment, subsystem, component, or hardware or software requirements, while ground segment requirements management is the process of establishing and maintaining ground segment, subsystem, component, and hardware and software item requirements; archiving past baselined versions of requirements, disseminating information about the requirements, and providing support for the requirements management system.

16.2 Definitions

Acquirer An organization that procures products for itself or another organization [1].

Acquisition team The acquisition team includes the acquirer personnel as well as Federally Funded Research and Development Center (FFRDC), Systems Engineering and Technical Assistance (SETA), and Systems Engineering and Integration (SE&I) contractor personnel.

Baseline A product or a set of products that has been formally reviewed and agreed on at a particular point in the item's lifecycle, which thereafter serves as the basis for further development, and which can be changed only through change control procedures. Also used as a verb, e.g., "to baseline a product or a set of products." (Adapted from [2])H

Build A version of the system or software that meets a specified subset of the requirements that the completed system or software will meet [1]. Contractors use various synonyms for the term "build," such as "increment," "release," "block," "iteration," "cycle," "drop," "spiral," and "sprint." In this chapter, the

term “build” will be used for software development iterations, and the term “increment” will be used for the ground segment development iterations that may include hardware, software, and facilities.

Capability The ability to complete a task or execute a course of action under specified conditions and level of performance [3].

CDRL item The term “Contract Data Requirements List” (CDRL) is used for the list of deliverable data in a contract. Data here refers to information, usually described in documents. An individual deliverable data product is called a CDRL item. The CDRL does not include the operational hardware and software to be delivered for the ground segment and installed in the operational facility.

Child requirement A requirement in a child specification that can be traced upward to one or more requirements in one or more specifications in the specification tree immediately above that child specification. A child requirement may also be a derived requirement. See ‘parent requirement’ [1].

Child specification A specification in the specification tree that is immediately below another specification in the specification tree. See ‘parent specification’. Child specifications contain child requirements [1].

Component An element in the architecture of a ground segment subsystem. Components may consist of hardware and/or software items or may be procured as a single entity.

Data item description An annotated outline of required contents of a product, usually a product that is on the contract data requirements list to be delivered to the acquirer.

Derived requirement A requirement that results from architecture or design decisions or from the user’s concept of operations (CONOPS) or the contractor’s ground segment operational concepts and that is not directly traceable to one or higher-level requirements. A derived requirement may be either a functional or non-functional requirement [1].

Element A generic term used in this chapter for an architectural, design, or implementation entity, such as a subsystem, component, hardware item, software item, class, object, hardware unit, or software unit.

End user A person who uses information produced by the hardware and/or software in a ground segment to perform the functions required by his/her position. An end user may interface directly with the hardware and/or software in a ground segment to obtain the needed data.

Functional Requirement A requirement that defines a specific behavior or function of the system or software. In general, functional requirements define what a system is supposed to do, while non-functional requirements define how a system is supposed to be [1].

Hardware item An aggregation of hardware that satisfies an end-use function and is designated for specification, interfacing, qualification testing, configuration management, or other purposes [1].

Increment See "build".

Iterative lifecycle model A lifecycle model, such as incremental or evolutionary with repeating and possibly overlapping development activities. Note: Iterative software development lifecycle models include agile development (Adapted from [1]).

Lifecycle model A project management framework providing a sequencing strategy and a disciplined approach to the structure and order of activities

Non-functional requirement A system or software requirement that specifies criteria that can be used to judge the operation of the system or software, rather than specific behaviors. Contrast with functional requirements that define specific behavior or functions. Performance requirements, including response times, throughput, hard time deadlines, and accuracies, are considered non-functional requirements. Other examples of non-functional requirements are specialty engineering requirements (e.g., security, safety, reliability, maintainability, availability, supportability, and human systems integration), computer resource margin requirements, and scalability and extensibility requirements (Adapted from [1]).

Operator A person who directly interacts with the hardware and/or software in a ground segment in order to perform the functions required by his/her position. An operator is sometimes called a "crewmember."

Parent requirement A requirement in a parent specification that can be traced downward to one or more requirements in specifications in the specification tree immediately below that parent specification. See 'child requirement' [1].

Parent specification A specification in the specification tree that is immediately above one or more other specifications in the specification tree. See 'child specification'. Parent specifications contain parent requirements [1].

Process A set of interrelated activities, which transform inputs into outputs, to achieve a given purpose [2].

Product Information, software, or hardware created, modified, or incorporated by means of systems engineering, integration and test, or software or hardware development activities. Examples include plans, requirements, architecture, design, code, hardware units, databases, test information, and manuals (Adapted from [2]).

Requirements allocation The process of assigning top-level functional or non-functional requirements, usually in a decomposed form, among the lower-level partitioned functions for accomplishment. Thus, ground segment requirements are allocated to the subsystems; subsystem requirements are allocated to components; and component requirements may be allocated to hardware and software items. (Adapted from clause 3.2508 of [INCOSE 2011])

Requirements management system A software tool providing a database of requirements that supports the requirements development and management activities.

Software Assurance The planned and systematic set of activities that ensures that software processes and products conform to requirements, standards, and procedures to help achieve:

- Trustworthiness—No exploitable vulnerabilities exist, either of malicious or unintentional origin, and
- Predictable Execution—Justifiable confidence that software, when executed, functions as intended [7].

Software engineering "The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software" [4].

Software item An aggregation of software that satisfies an end-use function and is designated for specification, interfacing, qualification testing, configuration management, and other purposes. Software items are selected based on tradeoffs among software function, size, host or target computer systems, developer, support strategies, plans for reuse, criticality, interface considerations, the need to be separately documented and controlled, and other factors. A software item is composed of one or more software units. A software item is sometimes called a computer software configuration item (CSCI) [1].

Specification tree A hierarchy diagram showing the dependency relationships of all of the specifications in the ground segment. Thus, for this handbook, it shows the ground segment specification at the top level, the subsystem specifications at the next level, the component specifications at the next level, and the hardware item and software item specifications at the bottom level. Because some subsystems may consist of components and/or hardware and

software items, hardware and software item specifications may be directly subordinate to subsystem specifications.

Subsystem An element in the architecture of a ground segment. Subsystems generally consist of components, hardware items, and/or software items.

Stakeholder A group or individual that is affected by, or is in some way accountable for, the outcome of a product (Adapted from [2]).

User A person who interacts with the hardware and/or software in a ground segment either directly or indirectly to perform their job or a task. Thus, the term user includes both operators and end users. Note that a person may be both an operator and a user.

Verification level The integration level at which a requirement is to be verified. For example, a software item requirement may need to be verified at the software, component, subsystem, or ground segment level. A common example of this is an interface requirement, which may be verified by the method of test using a simulator at the software level, but which then needs to be verified by the method of demonstration at the subsystem level using the actual interface.

Verification method One of four techniques used to fully or partially verify a requirement: Inspection (I), Analysis (A), Demonstration (D), and Test (T). These techniques are also called "qualification methods." These four techniques are defined as follows:

Inspection	A method used to determine characteristics by inspecting engineering documentation produced during product development, including both hardware and software documentation, or by inspection of the product itself to verify conformance with specified requirements. Inspection generally is nondestructive and consists of visual inspections or simple measurements without the use of precision measurement equipment.
Analysis	A method used to verify requirements by determining qualitative and quantitative properties and performance by studying and examining engineering drawings, software and hardware flow diagrams, software and hardware specifications, and other hardware and software documentation (e.g., commercial off-the-shelf vendor documentation), or by performing modeling, simulation, or calculations, or any combination, and analyzing the results. Similarity analysis is used in lieu of tests or demonstrations when it can be shown that an item is similar to, or identical in design to, another item that has been certified previously to equivalent or more stringent criteria.

Demonstration	A method used to verify requirements by exercising or operating the system or a part of the system in which instrumentation or special test equipment is not required beyond that inherently provided in the system being verified. In the demonstration method, sufficient data for requirements verification can be obtained by observing functional operation of the system or a part of the system. When this verification method generates data that are recorded by inherent instrumentation, inherent test equipment, or operational procedures, any analysis that must be performed using the data collected during the demonstration is an integral part of this method and should not be confused with the analysis method of verification described above.
Test	A method used to verify requirements by exercising or operating the system or a part of the system using instrumentation (hardware or software or both) or special test equipment that is not an integral part of the system being verified. The test method by its nature generates data, which are recorded by the instrumentation, test equipment, or procedures. Analysis or review is performed on the data derived from the testing. This analysis, as described here, is an integral part of this method and should not be confused with the analysis method of verification described above.

In some unusual cases, a unique verification method must be used. Such a method is generally known as Special (S). (Adapted from [1])

Waterfall lifecycle model A system or software lifecycle model in which the constituent activities are performed once in sequential order, possibly with overlap but with little or no iteration. For development, the activities typically include requirements development; architecture; detailed design; implementation/fabrication/procurement and unit testing; integration and testing; and qualification and verification testing (Adapted from [1]).

16.3 Ground Segment Requirements Development

Ground segment requirements development during the acquisition and development lifecycle phases includes the decomposition of the ground segment into subsystems, components, and hardware and software items, and the associated development of requirements through these decomposition levels.

16.3.1 Ground Segment Requirements Development through the Acquisition and Development Lifecycles

Figure 16-1 depicts the lifecycle phases from the *Mission Assurance Guide (MAG)* [8] aligned with the DOD acquisition lifecycle phases from DODI 5000.02 [9]. Ground segment requirements development is a full lifecycle activity that occurs throughout all of the acquisition lifecycle phases, although the predominant requirements development activity occurs during the early lifecycle phases (e.g., in MAG phases Concept Studies through Complete Design).

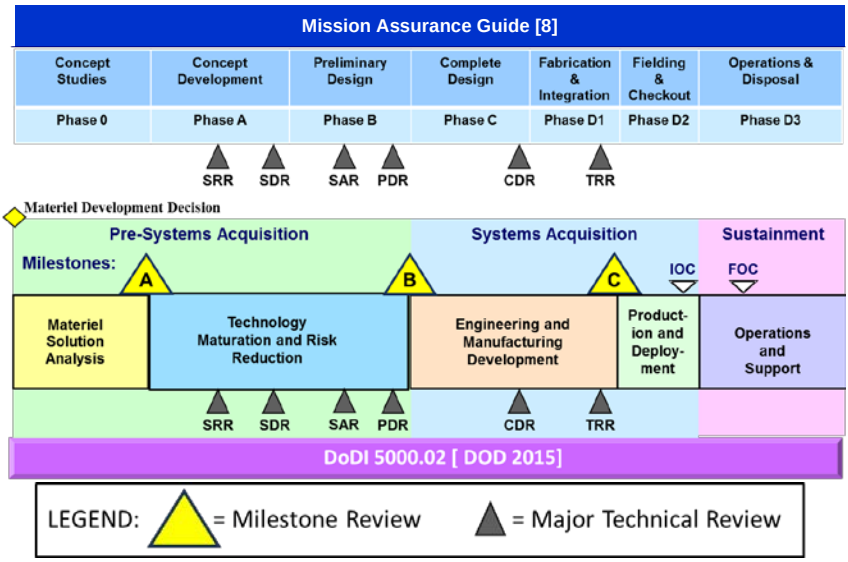


Figure 16-1. MAG and DOD lifecycle phases aligned.

Figure 16-2, the defense-unique software-intensive program model, shows an acquisition lifecycle model frequently used for ground segment acquisition. Figures 16-3 through 16-5 provide a brief description of ground segment requirements development activities in each of the acquisition phases in Figure 16-2. Other acquisition lifecycle models can be mapped into the DOD acquisition lifecycle models by aligning the major technical reviews and acquisition milestones.

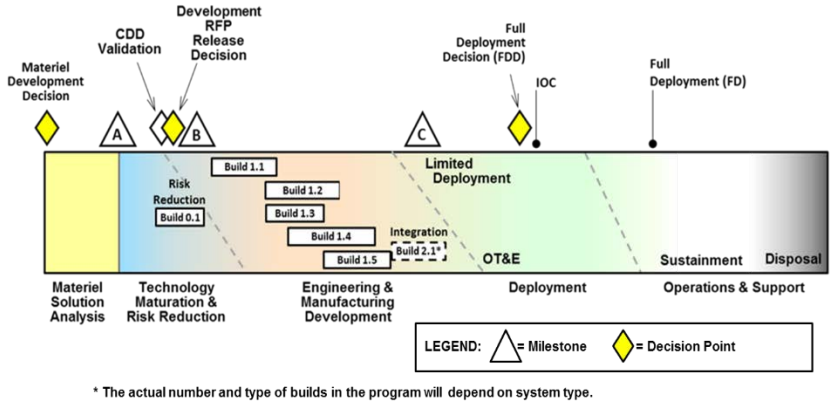


Figure 16-2. Model 2: Defense-unique software intensive program [9].

Before the ground segment acquisition begins, the eventual owner of the ground segment (usually the user organization) documents the capabilities that are needed in the new system. A capability is defined to be "the ability to complete a task or execute a course of action under specified conditions and level of performance" [3]. In the DOD environment, the user organization prepares an initial capabilities document (ICD) containing the needed capabilities that the user does not currently have. This document is completed and approved before the start of the acquisition lifecycle at the materiel development decision (MDD) milestone and is input to the materiel solution analysis (MSA) phase. (Other organizations document the user's needs differently, but new ground segment development or updates to existing ground segments are always based on the user's needs that are not being met by existing capabilities.)

During the MSA phase the acquirer performs studies of alternative architectures, technology maturation, modeling and simulation, prototype development, and other engineering analyses, either within the acquisition team itself or from contractors. This set of activities leads the acquirer to develop a preferred ground segment architecture and the technical requirements document (TRD). The TRD eventually becomes the contractual requirements specification for the ground segment being acquired. During the same time period the user organization is preparing the capabilities development document (CDD) that elaborates the ICD into more detailed capabilities that the user organization needs to be implemented in the ground segment. Ideally, the user and acquirer organizations cooperate during this phase so that the CDD and TRD are developed iteratively, using the results of all of the engineering analyses and studies performed during the MSA phase. MSA ground segment requirements development is depicted in Figure 16-3.

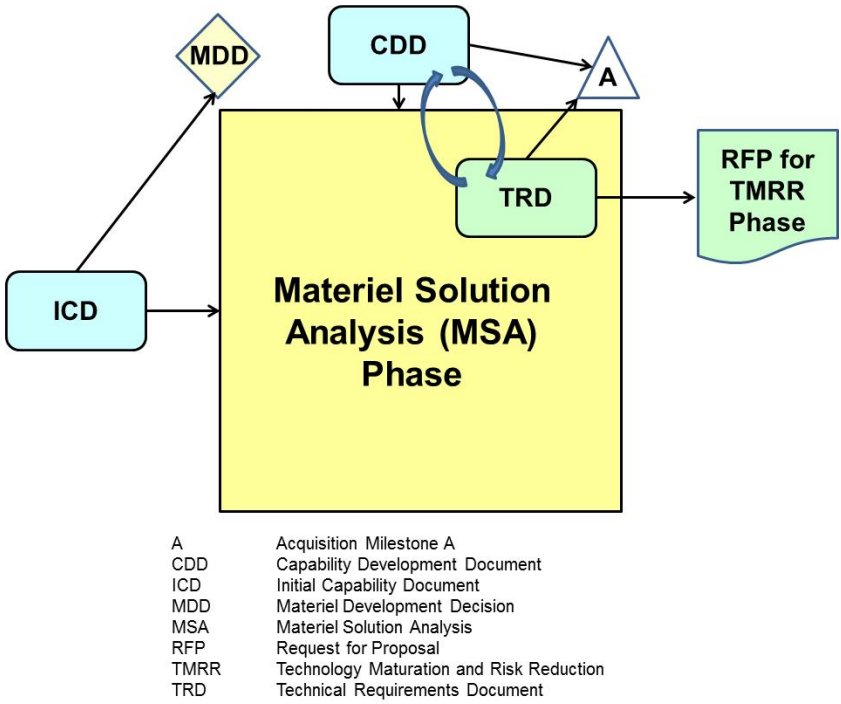


Figure 16-3. Requirements development in the MSA phase.

Development of the requirements for the ground segment being acquired is one of the major activities of the technology maturation and risk reduction (TMRR) phase. During this phase one or more contractors perform architecture studies, engineering analyses, modeling and simulation, prototype development, and technology maturation. In addition, the contractors develop the ground segment requirements, architecture, and design through the preliminary design phase, which culminates in the preliminary design review (PDR). For requirements, the contractors develop their ground segment specification, the ground segment subsystem and component specifications, and the hardware item and software item requirements specifications. The ground segment specification is usually reviewed at a system requirements review (SRR), and the subsystem and component specifications are usually reviewed at a system design review (SDR), also called a system functional review (SFR). The software item specifications are reviewed at the software requirements and architecture review (SAR), and the hardware item specifications are reviewed at the PDR. The contractors' ground segment requirements specifications must flow down from and satisfy the contractual TRD. The ground segment decomposition into subsystems, components, and hardware and software items.

During the TMRR phase the user and acquirer prepare an updated CDD and TRD, respectively. The updated CDD provides the user's needed capabilities to be provided by the ground segment against which they will test the implemented ground segment for acceptance from the acquirer. The updated TRD becomes the contractual requirements specification for the engineering and manufacturing development (EMD) contract under which the ground segment will be implemented. Ideally, the user and acquirer organizations cooperate so that the updated CDD and TRD are developed iteratively, using the results of the contractors' engineering work performed during the TMRR phase. Ground segment requirements development in the TMRR phase is depicted in Figure 16-4.

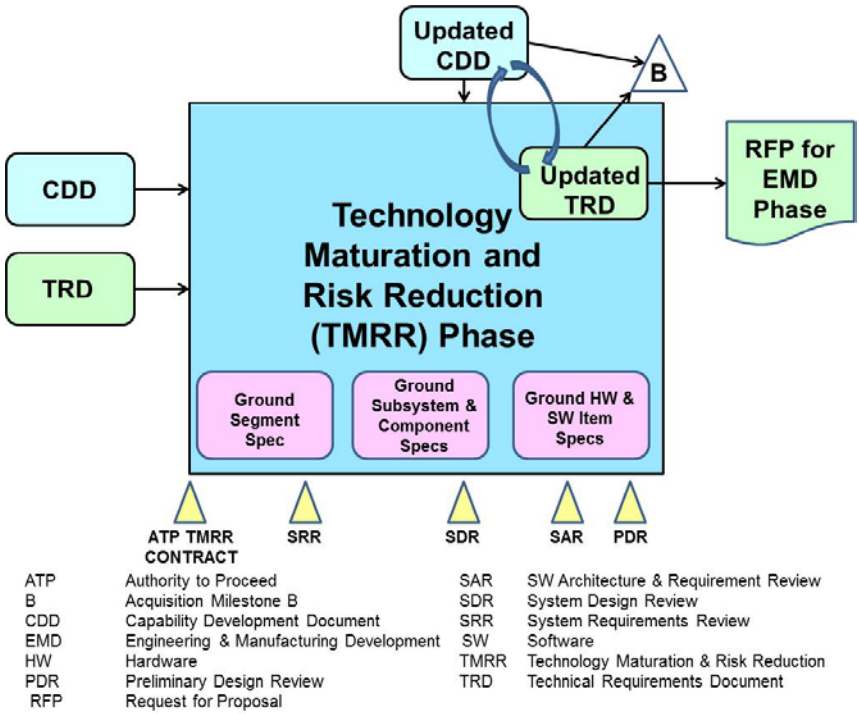


Figure 16-4. Requirements development in the TMRR phase.

During the EMD and deployment phases the contractor team selected to build the ground segment will update the ground segment, subsystem, component, and hardware and software item specifications developed during the TMRR phase. New and modified capabilities may have been incorporated into the user's updated CDD and reflected in the acquirer's TRD. Thus, the contractual TRD for the EMD and deployment contract may have differences from the TRD for the TMRR phase. Such changes usually necessitate changes to the specifications

developed by the contractor in the earlier phase. If the differences are significant, delta requirements and/or design reviews may be held to review the updated specifications. Following the CDR, requirements development continues in an update mode through the remainder of EMD and deployment, during which continuing changes to requirements at all levels are handled. Requirements update continues throughout the rest of the life of the ground segment during operations and support (O&S). Ground segment requirements development during EMD, deployment, and O&S is depicted in Figure 16-4.

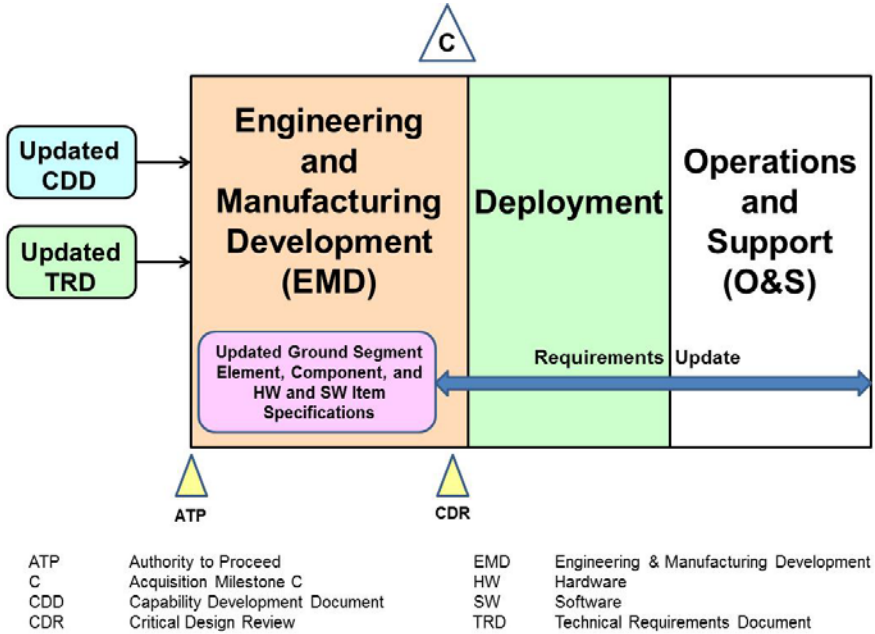


Figure 16-5. Requirements development during EMD, deployment, and O&S.

Because most ground segments are developed using iterative acquisition or development lifecycle models, ground segment requirements development is generally more complex. When an iterative lifecycle model is used, requirements will be developed on an iteration-by-iteration basis. If an iterative acquisition lifecycle model is used, such as that shown in Figure 16-2, the acquisition team develops a TRD for each ground segment increment because each increment is being acquired under a separate contract. Sometimes the user will have a separate CDD for each increment. However, frequently the CDD will address the final ground segment, and the acquisition team will need to decide how to allocate the user's capabilities to increments so that the ground segment will provide interim capabilities when needed by the user.

Even if the ground segment is not acquired iteratively, the contractor may decide to develop the ground segment in iterations, especially if the ground segment is large. In this case, the contractor decides which TRD requirements to allocate to each increment. The ground segment, subsystem, component, and hardware and software item requirements specifications will be developed in each increment for the requirements allocated to that increment. Sometimes the ground segment specification will be developed for the entire ground segment so that it satisfied all of the TRD requirements. The contractor will then allocate the ground segment specification requirements to their increments, and the lower-level requirements specifications will be developed on an increment-by-increment basis.

While hardware items are usually developed using a waterfall (once through) lifecycle model, software items are usually developed using an iterative lifecycle model on a build-by-build basis. Software item requirements may be developed prior to the beginning of development of the builds, or may be developed in each build for that build only.

Upgrades to existing ground segments frequently do not have a separate MSA phase and use a single contract to perform the requirements development, architecture, design, implementation, and deployment of the updated ground segment. The above discussion on ground segment requirements development in each phase is adaptable to this and other variations of the acquisition and development lifecycles.

If the system being acquired consists of both a space segment and a ground segment, the CDD and TRD will address the entire system, as will the contractors' system specification developed in the TMRR phase. There will then be an additional step where the contractor allocates the requirements to the two segments and develops the segment level specifications. At that point requirements development for the ground segment proceeds as described.

16.3.2 Ground Segment Requirements Decomposition

The activities of requirements development and architecture definition are usually performed concurrently and iteratively. The architecture definition decomposes the ground segment into its constituent subsystems, components, and hardware and software items. In this handbook, the ground segment is considered to consist of a set of subsystems each of which consists of a set of components, hardware items, and/or software items. The components may themselves consist of hardware and software items, or they may be single entities (e.g., a commercial off-the-shelf [COTS] front-end communication protocol handling system).

At each level of architectural decomposition, the requirements are allocated to the ground segment architectural elements at that level. To develop the requirements specifications for the architectural elements at each level, requirements allocated to those architectural elements are elaborated from their parent requirements, and derived requirements based on architecture and design decisions and operational concepts are added. This process of architecture decomposition and requirements development continues until the ground segment is fully decomposed into the components, hardware items, and software items that will be developed or procured.

Requirements for each of the ground segment subsystems must be developed and must satisfy the requirements allocated to that subsystem from the ground segment specification. Figure 16-6 illustrates the major subsystems and shows components for some of these subsystems. For example, the space/ground asset command and control subsystem is shown to have the following components:

- Telemetry processing
- Satellite command and control
- Ground asset command and control
- Anomaly detection and resolution

The space/ground command and control subsystem requirements are allocated to each of these four components, following which the component requirements are developed. This process repeats as each component is divided into the hardware items and software items that must be implemented or procured.

An example ground segment specification tree based on the reference architecture is shown in Figure 16-7. A specification tree is a hierarchical diagram that shows the parent and child specifications for each layer in the architecture. The figure shows the ground segment requirements specification at the top. The next layer shows the specifications for each of the subsystems in the reference architecture. Because of space limitation in the figure, the lower levels are shown for only one subsystem, the space/ground asset command and control subsystem. This subsystem has four components in the reference architecture, leading to the four component specifications shown in Figure 16-7. Each of these four components may consist of some number of hardware and software items. If a component is to be procured as an entity, that component is not usually decomposed in the architecture, and the component specification becomes the basis for a procurement specification developed later in the design phase. A specification tree frequently contains the specification identifiers in each of the boxes. Such identifiers are assigned by the configuration management process.

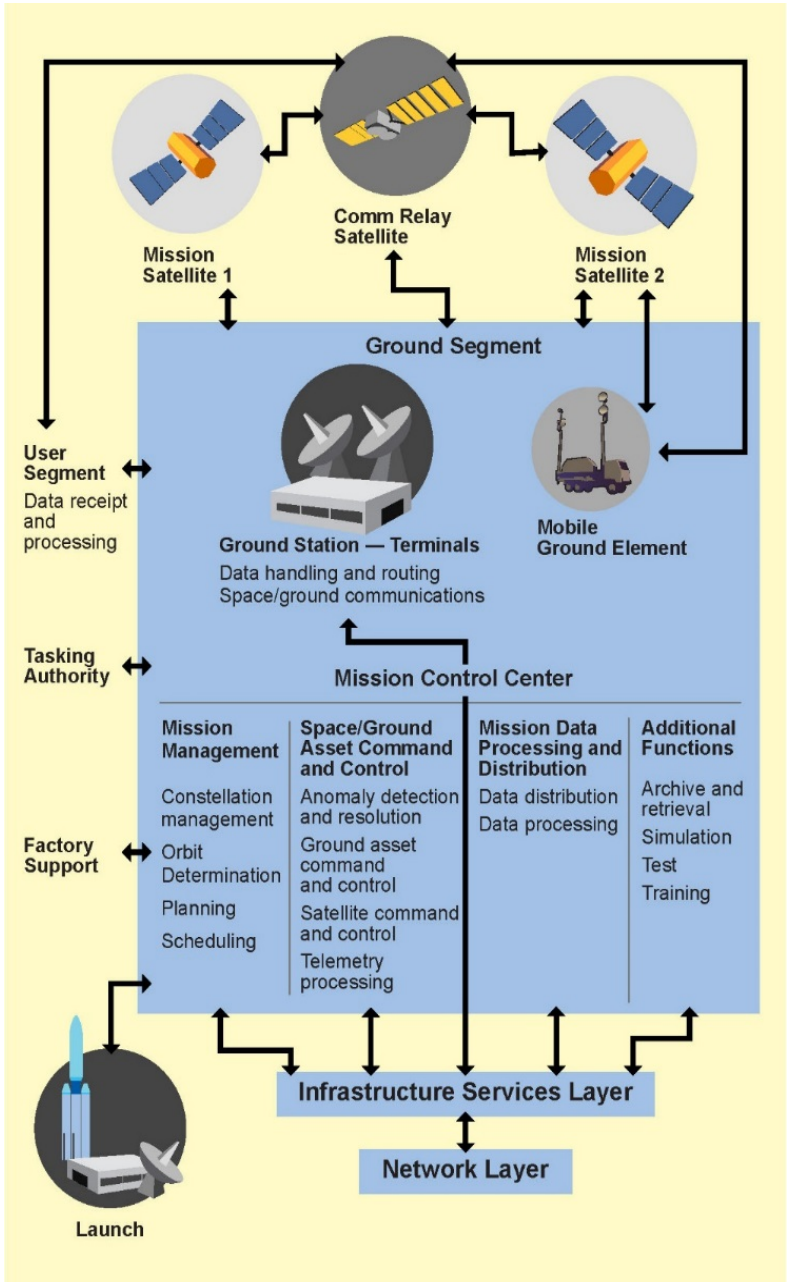


Figure 16-6. Ground segment reference architecture.

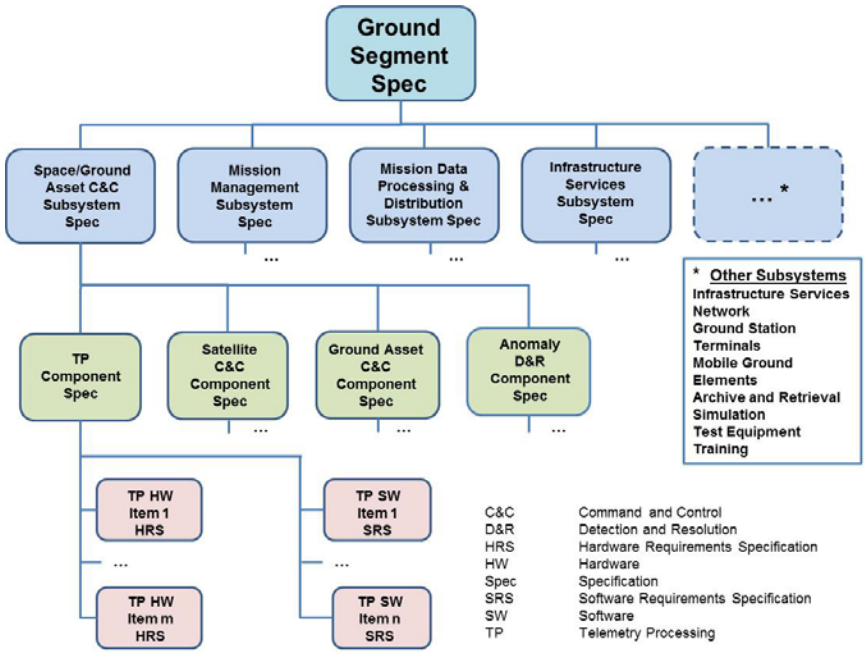


Figure 16-7. Example ground segment specification tree.

16.3.3 Types of Ground Segment Requirements

Ground segment requirements can be considered as consisting of four types: functional, non-functional, interface, and qualification. A functional requirement is one that defines a specific behavior or function of the ground segment or its subsystems, components, hardware items, or software items. A non-functional requirement is a characteristic that can be used to judge the operation of the ground segment or its constituent elements. In general, functional and interface requirements define what a ground segment is supposed to 'do', while non-functional requirements define how a ground segment is supposed to 'be'. An interface requirement defines the data required to be sent or received across interfaces with other entities, including external systems and other elements of the ground segment, and the physical characteristics of these interfaces and the data (e.g., formats, data rates). A qualification requirement defines the method(s) to be used to verify that a functional, non-functional, or interface requirement is satisfied by the implemented ground segment element. Qualification requirements are also known as verification requirements.

The non-functional requirements are equally as important as the functional and interface requirements, because satisfaction of non-functional requirements is what makes a ground segment usable during operations. Performance

requirements, including response times, throughput, timing deadlines, and accuracies, are non-functional requirements. Other examples of non-functional requirements important to ground segments are as follows:

- Operational life requirements
- Specialty engineering requirements
  - Security, including cybersecurity
  - Safety
  - Reliability, maintainability, and availability (RMA)
  - Supportability, including fault management and fault tolerance
  - Human systems integration
- Computer resource margin requirements
- Scalability and extensibility requirements
- Software engineering requirements, including software assurance requirements
- Interoperability and open systems requirements
- Usability requirements, including automation and autonomy requirements
- Environmental requirements (e.g., temperature and altitude ranges of operations)
- Survivability requirements
- Design and construction constraints (e.g., weight, footprint, and power limitations)
- Legacy system requirements (e.g., backwards compatibility)

The user's CDD will frequently specify only functional, performance, availability, operational life requirements, and organizational interfaces. The acquisition team must derive a full set of ground segment requirements, including functional, non-functional, interface, and qualification requirements for the TRD. Anything not specified in the ground segment TRD is part of the contractor's engineering trade space, so it is very important for the acquirer to specify exactly what is needed for the ground segment to meet the CDD and function properly in the expected operational environment.

As the contractor develops their ground segment requirements specification, the non-functional requirements must be flowed down from the TRD and elaborated as needed, just as must be done with the TRD functional and interface requirements. As the ground segment requirements are allocated to the subsystems, the non-functional requirements must be allocated as well. Each ground segment quantitative requirement, such as performance requirements and RMA, must be met by the joint set of its allocated quantitative requirements.

At each level of the specification tree, the acquisition team should thoroughly review the requirements specifications to ensure functional, non-functional,

interface, and qualification requirements have been completely and accurately specified.

16.3.4 Ground Segment Requirements Lessons Learned

The following lessons learned apply to ground segment requirements development:

- The contractual TRD is an extremely important document, equal in importance to the RFP. As such, it should have a broad participation by the acquisition team in its development. This includes experts from all disciplines within the acquisition team, including systems engineering, system test, specialty engineering, and hardware and software personnel.
- In absence of a required set of contents for the TRD, which may be unique to acquisition organizations or programs, the content of the System/Subsystem Specification (SSS) Data Item Description (DID) [10] may be used as a starting place.
- The contractual TRD should have a broad review including members of the acquisition team, users and other external stakeholders, potential bidding contractors, and acquisition management personnel.
- For the case where the acquisition covers both a space and ground segment, the TRD must not focus only on the space segment. The acquisition team must ensure that the contractual requirements for the ground segment are adequately included.
- The contractor's ground segment specification should be a required contractual delivery, i.e., a Contract Data Requirements List (CDRL) item. This CDRL item should always be required to be approved by the acquirer. Suggested content for this CDRL item is found in the SSS DID [10].
- Frequently the contract is changed following the approval of the ground segment specification by the acquirer so that the contractor's ground segment specification becomes the contractual requirements specification instead of the TRD. This works well as long as the contractor's ground segment specification has been thoroughly reviewed by the acquisition team and all other stakeholders and updated per their comments.
- It is strongly recommended that the subsystem requirement specifications be CDRL items, and that they be required to be approved by the acquirer. The same DID may be used for the subsystem specification as for the ground segment specification.
- The component requirements specifications are also recommended to be required to be CDRL items and approved by the acquirer. The same

DID may be used for the component requirements specifications as for the subsystem specifications.

- Because of the elevated risk in software development for ground segments, the software requirements specifications and interface requirements specifications (which may be used to specify the software interface requirements) are strongly recommended to be CDRL items and to be approved by the acquirer. Suggested content for these CDRL items is found in the software requirements specification (SRS) DID [11] and the interface requirements specification (IRS) DID [12].
- It is recommended that the hardware requirements specifications be required CDRL items for at least those hardware items being developed rather than procured, those hardware items of higher risk, and those hardware items considered critical.
- To ensure the contractors follow good systems engineering practices, a robust systems engineering standard, such as [13] or [14], should be placed on contract as a compliance document, and adherence to that standard should be enforced by the acquirer throughout the contract period.
- The major technical reviews should be placed on contract by requiring a robust standard such as from Peresztegy and the IEEE [15, 16, 17].
- Additional information on RFP development may be found in [18] and [7]. Additional information on recommended CDRL items may be found in work by Owens [19, 20].
- The contractor's ground segment specification must be thoroughly reviewed by the acquisition team. The review must be broad-based, including acquisition team members from all the various disciplines: ground segment systems engineering, ground segment testing, specialty engineering (e.g., RMA; security, including cybersecurity; safety; human systems integration; supportability; and survivability), various hardware disciplines, and various software disciplines. The review must include all stakeholders, including end users, operators, acquisition management, legal, procurement, and personnel from interfacing systems. Finally, the reviewers must include domain specialists (e.g., mission data processing, mission management, space and ground asset command and control, simulation, networks, ground-ground and space-ground communications, and infrastructure).
- The lower-level ground segment specifications (i.e., for subsystems, components, and hardware and software items) must also be thoroughly reviewed by the acquisition team. Specialists from the applicable engineering disciplines and domains must be included in these reviews.

16.4 Ground Segment Requirements Engineering

Requirements engineering is a systems engineering discipline that consists of requirements development and requirements management.

16.4.1 Ground Segment Requirements Development Process and Activities

The process of defining and documenting ground segment, subsystem, component, or hardware or software requirements. Requirements development consists of requirements elicitation, analysis, decomposition/allocation, elaboration, derivation, review, verification, and validation.

Requirements development is a rigorous process with a purpose to establish complete, consistent, correct, current, clearly defined, and verifiable requirements. It is performed by the acquirer to prepare the TRD, and by the contractor to prepare the ground segment requirements specification and all of the lower-level specifications in the ground segment specification tree.

The ground segment requirements development process is shown in Figure 16-5. As shown in the figure, requirements development is an iterative process that should provide successive refinement of the requirements through all levels of the requirements baseline and throughout the entire ground segment lifecycle.

Requirements development consists of the following activities, which are performed iteratively and concurrently:

- Requirements elicitation
- Requirements analysis
- Requirements decomposition/allocation
- Requirements elaboration
- Requirements derivation
- Requirements review
- Requirements verification
- Requirements validation

Figure 16-8 shows the basic requirements development process in the center. Requirements review, verification, and validation are shown at the center of the iteration of requirements development activities because they apply to all of the other activities. The top of the figure shows the context in which the requirements are being developed. This context includes the applicable government regulations, instructions, and guidance; specifications and standards; requirements development processes, techniques, and tools; and requirements development and domain best practices.

The requirements development activities occur at each level of the government and contractor ground segment requirements baseline. Thus, the inputs to the iterative requirements development activities include all requirements specifications at a higher level and the outputs include the requirements specifications for the particular level being developed. Input to the requirements development process includes architecture and design decisions and concepts of operation, which guide the derivation of requirements. For example, for the preparation of the acquirer's TRD, the input would be the user's ICD and CDD and CONOPS, the government reference architecture, and other design decisions (e.g., based on technology maturity). For the preparation of the contractor's ground segment system specification, the input would be the TRD, architecture and design decisions from the TMRR phase studies, and the contractor's operational concept for how the ground segment would operate to fulfill the user's CONOPS. The lower-level requirements specifications (i.e., subsystem, component, hardware item, and software item) specifications similarly would have the parent requirements specifications, architecture and design decisions applicable to their level, and the ground segment operational concepts as input. Interface requirements specifications are shown as an output because they may be documented separately.

There is no industry standard for the definition of the requirements development activities. Thus, contractors may use different terminology than that presented here. This is acceptable as long as they include all of the types of requirements development activities described below, even if the activities are called by different names.

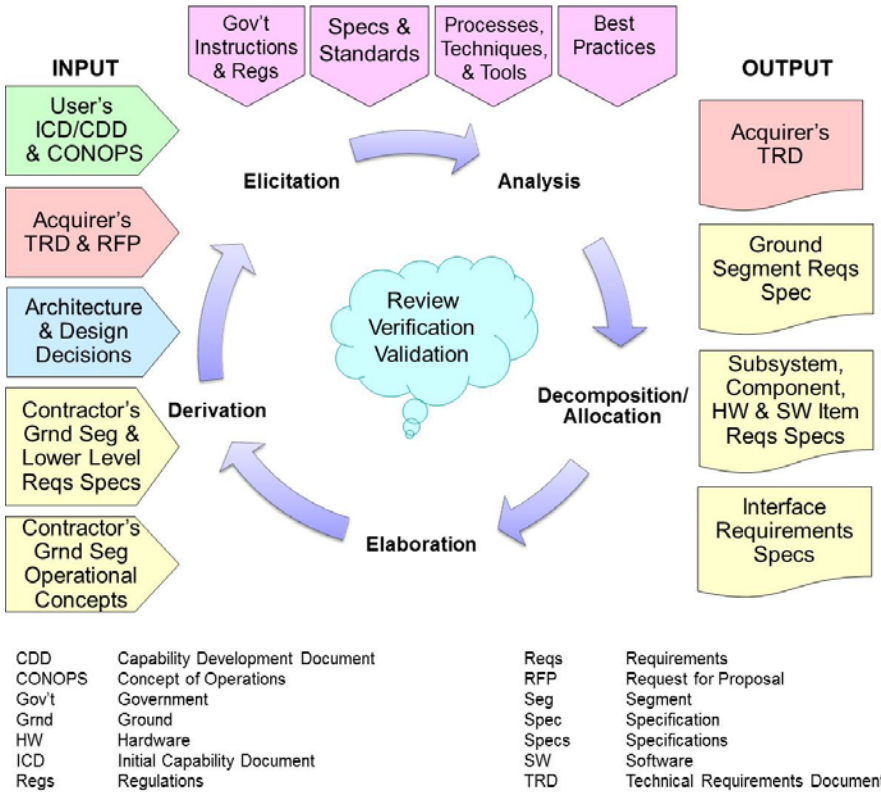


Figure 16-8. Ground segment requirements development process.

Requirements Elicitation

Requirements elicitation is the process through which the needs of the users, including end users and operators, and other stakeholders are discovered and then reviewed, articulated, understood, and documented. Requirements elicitation may use systematic techniques such as prototypes and structured surveys to proactively identify and document the ground segment end user and operator needs. (Adapted from clause 3.2512 of [5].

Requirements elicitation should occur at all levels of the specification tree, not simply at the TRD or ground segment specification level. For the TRD and ground segment specification, stakeholders other than users include:

- Ground segment systems engineers, system architects, and system testers
- Specialty engineering personnel (e.g., security, cybersecurity, safety, RMA, human systems integration, supportability, and survivability)

- Representatives of interfacing external systems
- Representatives of interfacing segments (e.g., space, launch, and user segments)
- Ground segment hardware and software engineers with applicable domain and discipline expertise

At lower levels of the ground segment specification tree, stakeholders also include architects, designers, implementers, testers, and specialty engineers for the ground segment subsystems, components, and/or hardware and software items undergoing requirements development, as well as applicable personnel from interfacing subsystems, components, and/or hardware and software items.

Requirements elicitation may use systematic techniques to proactively identify and document the ground segment end-user and operator needs. Potential types of techniques include simulations, prototypes, mockups of displays and/or reports, day-in-the-life analyses, scenarios and/or use cases, and targeted surveys. Requirements developers should also observe current operations of legacy systems, if applicable, so that they can understand the shortcoming of these systems and the necessary requirements of the new system. Examples of such systems are satellite command and control systems for satellites on orbit, mission command and control systems for mission payloads aboard existing satellites, and ground control of existing world-wide ground segment assets. The understanding obtained by observing operations is invaluable for discussing the shortcomings and needed upgrades with the operators and end users. Requirements developers should also review the specifications of the legacy systems to identify functionality and potential requirements, and should review the CONOPS with the users to ensure they understand the users' ideas of how the new ground segment should operate.

Requirements elicitation will frequently result in conflicting requirements from different users and stakeholders and may also result in requirements that are not supported by the ground segment acquisition's cost and schedule budgets. The job of the requirements developers includes adjudication of conflicting requirements and deciding upon which of the conflicting requirements will be in the requirements baseline. The resulting set of requirements must also be consistent with the allocated cost and schedule budgets. For the TRD, it is the job of the acquirer's requirements developers to specify a set of requirements that is consistent with the program's acquisition performance baseline (i.e., the acquisition program manager's requirements, cost, and schedule baseline). For the ground segment system specification, it is the job of the contractor's requirements developers to specify a set of requirements that is consistent with the contractual requirements, cost, and schedule baseline. To accomplish this, it is usually necessary to prioritize the requirements so that critical, necessary, and desired capabilities are identified and included according to the applicable cost and schedule budget. Similarly, lower-level requirements must be consistent

with the cost and schedule budgets allocated to each element of the ground segment architecture in the specification tree.

The resulting set of elicited requirements should be reviewed with the users and other stakeholders to identify issues and gaps before the next stage of requirements development. This review can also help users and other stakeholders understand what requirements can and cannot be met within the allocated cost and schedule baseline; for example, the degree of automation that can be afforded in the ground segment being developed.

Requirements Analysis

Requirements analysis is the process of using mathematical analyses, engineering models, prototypes, simulations, and other techniques, including engineering judgement, to systematically investigate requirements. Of particular importance is the analysis to support specification and decomposition of quantitative non-functional ground segment requirements, such as performance, timing, capacity, accuracy, RMA, computer resource margins, and other types of quantitative requirements.

Requirements analysis helps to provide a rigorous basis for the development of requirements and supports all of the other requirements development activities. Requirements analysis is used to determine the correctness and feasibility of requirements, based on user needs, physical environments, ground segment architecture decisions, and other ground segment constraints.

Requirements Decomposition/Allocation

Requirements decomposition and requirements allocation are closely related. Both are involved in the process of flowing down requirements from one level of the specification tree to the next.

Requirements decomposition is the process of separating or apportioning a requirement from a parent specification into a set of requirements for the child specifications at the next level of the specification tree. Thus, a ground segment requirement that applies to two subsystems will be decomposed into two requirements, one for each subsystem. A ground segment requirement that applies to multiple subsystems will be decomposed into a requirement for each applicable subsystem, and similarly for components, hardware items, and software items as the process continues down through the specification tree. Sometimes a parent requirement applies to only one child specification and so does not need decomposition. It is also possible that a parent requirement applies in its identical wording to all subordinate requirements specifications. An example of this might be a ground segment human systems integration requirement (e.g., that a specific standard be used for displays) that may apply in the same exact wording to all subsystems that will eventually have displays.

The decomposition of non-functional requirements, especially the quantitative requirements, is generally a complex task. Analysis is usually needed to apportion the quantitative requirements correctly. Almost never is a simple division of a quantitative requirement by the number of next level specifications correct. As example, the apportionment of quantitative RMA requirements usually needs a reliability model or models, and the apportionment of performance requirements may need a simulation model to obtain the correct allocation of the requirements into their constituent requirements at the next level.

The process of assigning top-level functional or non-functional requirements, usually in a decomposed form, among the lower-level partitioned functions for accomplishment. Thus, ground segment requirements are allocated to the subsystems; subsystem requirements are allocated to components; and component requirements may be allocated to hardware and software items. (Adapted from clause 3.2508 of [5]). All types of ground segment requirements, both functional and non-functional, in each level of the specification tree must be decomposed as needed and allocated to the elements in the next lower level of the specification tree. (See Section 16.3.3 earlier for a description of types of requirements.)

Requirements Elaboration

Requirements elaboration is the process of expanding an allocated requirement into a set of requirements that will, when implemented, allow the allocated requirement to be met by the element to which that requirement was allocated. Requirements from the ground segment specification allocated to subsystems will commonly need to be elaborated into multiple requirements at the subsystem level; subsystem requirements allocated to components will commonly need to be elaborated into multiple requirements at the component level; and so forth down through the hardware and software items.

The elaboration of allocated requirements is one of the most important steps in the requirements development process, but is frequently ignored by requirements developers. There are very few parent requirements that will not need elaboration in their child specifications because the requirements in any specification must be written at the correct level for that specification, that is, must be written so that the responsible engineers for the child specifications know what to implement in their elements.

As an example, a ground segment requirement to process the telemetry for link A of legacy satellites will have that requirement allocated to the space/ground asset command and control subsystem (see Figure 18-4). At the subsystem level this requirement will need to be elaborated into the various functional and performance requirements needed to process that link of telemetry, expanding (for example) into requirements for telemetry capture, decommutation,

processing and analysis, alarm/event processing, displays, trending, anomaly detection and resolution, and command and control of ground assets needed for processing this telemetry (e.g., antennas, communication equipment, COTS telemetry processing systems, computer servers, and operator stations). These requirements at the subsystem level are then decomposed and allocated, for example, to the components shown in Figure 18-4. The allocated requirements are then elaborated to component-level requirements for each component, and so forth down through the hardware and software items in the ground segment specification tree. At the lowest level, the requirements for the hardware and software items must be detailed enough for the hardware and software engineers to be able to architect, design, implement, and test the hardware and software items. For components to be procured as a COTS product, the requirements in the component level specification must be detailed enough for the engineers to produce a procurement specification for the exact COTS product needed.

Requirements Derivation

Requirements derivation is the process of defining requirements that are not directly traceable to a parent requirement, but rather are necessary due to architecture or design decisions, trade studies, or to either the user's CONOPS or the contractor's ground segment operational concepts. Derived requirements may also be specified based on knowledge about the domain or discipline or on best engineering practices. A derived requirement may be either a functional or non-functional requirement.

Some contractor processes require that all requirements be traceable to one or more parent requirements. In this case, the derived requirements are usually traced to a general top-level requirement. It is better, however, to trace the derived requirements to the source of the decision to include the requirement, for example to an engineering memo, architecture or design documentation, or operational concepts document.

Requirements Review

Requirements review is the process of reviewing the requirements for completeness, consistency, clarity, verifiability, traceability, feasibility, and other criteria. Requirements reviews include requirements development team reviews, peer reviews, walkthroughs, product evaluations, joint reviews with acquisition team and contractor team personnel, and major technical reviews (e.g., SRR, SDR, SAR, PRD, and CDR). Requirements reviews should occur throughout the requirements development process to ensure that the specified requirements are of high quality (i.e., have little to no defects). Reviewers should range from a group of requirements development peers to others involved in the requirements development process; ground systems engineers and system testers; specialty engineering experts; subsystem, component and hardware and software item engineers; users; acquirers; and other stakeholders

within the contractor team and government. Requirements reviews should always be performed using a checklist of criteria.

Requirements Verification

Requirements verification is the process of confirming that the specified requirements (both individually and collectively) are correctly and completely defined. While requirements verification also means the activity of ensuring the implemented ground segment, subsystems, components, hardware items, and software items properly satisfy their specified requirements, here requirements verification focuses on examination of the ground segment, subsystem, component, and hardware item and software item requirements to ensure their correctness and completeness. (Adapted from [1] and clause 4.1.23 of [6])

There are a number of techniques used for requirements verification, the most effective of which is peer review. Peer review is a detailed technical review of a set of requirements by a knowledgeable group of peers, usually using a documented set of criteria against which the requirements are reviewed. In peer review the set of requirements under review is usually a functionally cohesive set of requirements from a single requirements specification that is small enough to be reviewed in a reasonably short period of time (e.g., one to two hours). Identified defects are documented and tracked to closure. Peer reviews may be held after any requirements development activity to review the set of requirements resulting from that activity, but should be held on all requirements in the completed versions of each requirements specification. The review of a set of requirements for a particular requirements specification that is performed by knowledgeable peers during the requirements development process to identify defects for removal (Adapted from [2]).

Other techniques for verification include walkthroughs and product evaluations. The set of reviewers for these techniques is broader than just the peers who are knowledgeable of a particular set of requirements and may include management, requirements developers for other specifications, algorithm developers, testers, specialty engineers, and hardware and software experts in various disciplines and domains.

Full requirements verification usually needs both peer reviews and walkthroughs or product evaluations to ensure that the requirements are reviewed from multiple viewpoints. For contractor-developed specifications, some of the requirements verification reviews should include the acquisition team. Some contractors maintain a strict definition of “peer” for the peer reviews and do not allow customers to participate. In this case there need to be other requirements verification reviews in which the acquisition team participates.

Requirements Validation
Requirements validation is the process of confirming that the specified requirements (both individually and collectively) will fulfill their intended use in the eventual operational environment when implemented. Requirements validation applies to all levels of requirements, i.e., TRD, ground segment, subsystem, component, and hardware and software item requirements. Validation ensures the "right" system is built, as intended by the users and other stakeholders (Adapted from [1] and clause 4.1.22 of [6].

Requirements validation includes user and stakeholder reviews similar to the walkthroughs and product evaluations used for requirements verification. However, requirements validation may also use other techniques, such as prototypes and simulations, to ensure that when the product specified by its requirements is developed, it fulfills its intended use in the operational environment. In any case, the users and other operational stakeholders should participate as reviewers in the requirements validation activities.

16.4.1.1 Requirements Review Criteria

All requirements should be reviewed against an established set of criteria to ensure that they are properly specified. The following list provides a set of requirements review criteria to which other criteria specific to the program, domain, or discipline should be added. There is no standard language for naming requirements evaluation criteria, so the contractor's processes and checklists may use different terminology than that given here.

- Contains All Applicable Information—The requirements in a specification address all items in the specified content of the specification, as required by the contract, CDRL item DID with contract tailoring, Statement of Work (SOW), and contractor processes. Completeness of the TRD means that the TRD addresses all content required by the acquisition organization or program. If there is no guiding requirements from the acquisition environment, completeness means that the TRD should address all content in the SSS DID (see Section 16.3.4 above).
- Completeness—All requirements essential to the correct implementation of the ground segment, subsystem, component, hardware item, or software item are included. Completeness also means that all necessary derived requirements are included. This includes requirements derived from architecture or design decisions, trade studies, domain or discipline best practices, or to either the user's CONOPS or the contractor's ground segment operational concepts.
- Correctness—There are no obvious errors in the requirements statements, either technical or editorial.

- Internal Consistency—Within each specification:
  - No two requirements or other statements in the specification contradict one another.
  - A given term, acronym, or abbreviation means the same thing throughout the specification.
  - A given item or concept is referred to by the same name or description throughout the specification.
- Consistency Across Specifications—Within the specifications in the ground segment's specification tree:
  - No requirement or other statement in one specification contradicts a requirement or other statement in the other specifications in the specification tree.
  - A given term, acronym, or abbreviation means the same thing in all of the specifications in the specification tree.
  - A given item or concept is referred to by the same name or description in all of the specifications in the specification tree.
- Consistency with operational concepts—The requirements, both individually and collectively, must be consistent with both the user's CONOPS and the contractor's ground segment operational concepts. Necessary requirements derived from these documents must be included.
- Traceability—There is a current, correct, and easily accessible record documenting the relationship between:
  - Each requirement in each parent specification and the requirements in the child specifications whose source is the parent requirement. Note that for the Ground Segment Specification, the parent specification is the TRD, and for the TRD, the parent is the CDD.
  - Each requirement in each child specification and the requirements in one or more parent specifications which are the sources for the child requirement. Derived child requirements may be traced to a general higher-level requirement or to the documented source of the derived requirement (e.g., architecture or design decisions, operational concepts, domain or discipline knowledge, or best engineering practices).

  Note that this relationship is bi-directional and may be many-to-many.

- Coverage—For each requirement in each parent specification, the set of child requirements traced to the parent requirement, together with applicable derived requirements, will result in the satisfaction of the parent requirement when the ground segment child elements are implemented and their requirements verified. Another way to state this is that the set of child requirements, perhaps with applicable derived requirements included, fully cover the parent requirements.

- Concise—Each requirements statement must contain one and only one requirement. Multiple requirements should be broken out into separate sentences even if doing so causes repetitive language. In addition, each requirement should have a unique identifier assigned to it.
- Verifiability—A requirement is verifiable if one or more objective and feasible tests can be designed that together will fully determine whether the requirement has been met. These tests use one or more of the four verification methods specified in the requirements specification: Inspection (I), Analysis (A), Demonstration (D), and Test (T). Occasionally a requirement will need a Special (S) verification method. The qualification requirements in each specification document a required verification method to ensure each requirement is met by the implemented element. The criterion of verifiability includes checking that each requirements specification includes one or more appropriate verification methods for each requirement in the specification and that the specified verification method(s) for each requirement will enable the requirement to be verified using the method(s).
- Clarity: The requirement is understandable and unambiguous to the intended audience. This criterion includes avoiding the use of ambiguous words, such as "minimize," "maximize," "optimize," "adequate," "rapid," "user-friendly," "support," and "sufficient."
- Feasibility—Based on engineering knowledge and experience, the requirement violates no known principles or lessons learned that would render it impossible, extremely difficult, or of very high risk to develop or test, within cost and schedule constraints.
- Level of detail—The level of detail in a requirements specification is sufficient for domain and discipline experts to be able to architect, design, and implement the element addressed by the specification. The requirements must specify what the ground segment element must do, but not how it will be done. That is, the level of detail of the requirements must not unnecessarily constrain the architecture and design, design margins, or the trade space.
- Use appropriate requirements language—Requirements must be stated to clearly differentiate between requirements and non-requirements. In particular,
    - Requirements containing "shall" are mandatory and must be satisfied by the implemented ground segment, subsystem, component, hardware item, or software item.
    - Requirements containing "should" indicate goals that are desired to be met in the implemented ground segment element but are not mandatory.
    - Requirements containing "may" indicate a non-mandatory option.
    - The present and future tenses (e.g., "is" and "will") must never be used for requirements. The present tense should be used only for explanatory tense, and the future text only for government intent.

- Proper use of TBDs, TBRs, and TBSs—These indicators are an important measure of the maturity of the requirements and serve as placeholders for requirements that need to be explored further. If remaining in the TRD as placed on contract, they reflect some trade space being left to the contractor or areas where there is not enough information to fully define the requirement at the time the document in released. The meanings of these terms are as follows:
  - TBD—To be determined
  - TBR—To be resolved or to be revised
  - TBS—To be specified or to be supplied
- Proper grammar—Proper grammar in the statement of requirements is essential to their correct interpretation and implementation in the ground segment element. In particular, there are grammatical constructions that must be avoided, including the following:
  - Ambiguous modifiers where the subject of the modifier has more than one possibility.
  - Negative predicates should never be used. It is impossible to verify a requirement that is stated in the negative. Always state requirements in the positive.
  - Parenthetical expressions in requirements statement should never be used. The contents of any parenthetical are not legally part of the mandatory requirement, and thus may be ignored by the contractor.
  - Slashes should never be used in a requirements statement. Legally, slashes are interpreted as "or", not as "and".
  - Always use the active voice when stating requirements, for example "The ground segment shall…".
- Meets all contract requirements—The requirements must individually and collectively meet all of the contractual requirements, including statement of work, standards and their tailoring, CDRL item DIDs and tailoring on the CDRL item form, and any attachments to the contract such as special instructions for security and use of legacy hardware and software.

16.4.1.2 Ground Segment Requirements Management Process and Activities

The process of establishing and maintaining ground segment, subsystem, component, and hardware and software item requirements; archiving past baselined versions of requirements; disseminating information about the requirements; and providing support for the requirements management system. The purposes of the ground segment requirements management process are to support requirements development, enable rigorous control of changes to baselined requirements, and communicate requirements and their changes across the contractor and acquisition teams throughout the ground segment lifecycle.

The requirements management process is depicted in Figure 18-6. The figure shows that at the heart of the requirements management process is the database of requirements. Except for extremely small ground segment development projects, proper management of requirements necessitates a requirements management system (RMS) that operates on a central requirements database accessible by the entire requirements development and requirements management teams.

The RMS must be able to maintain (1) working areas for the requirements development team when they are developing new requirements or updating previously defined requirements, (2) current baselined versions of the requirements, and (3) archived versions of previous baselined projects. The RMS must also be able to handle all specifications in the ground segment specification tree along with the requirements contained in the specifications.

The requirements development team uses the requirements management system to support their activities. The RMS must allow the requirements development team full read and write access to the development versions of the specifications under development or update. The RMS must also allow them read access to the current baselined versions of the requirements and the archived versions of the requirements. A capable ground segment RMS should be able to perform the following additional types of functions:

- Provide unique identification for each requirement in all of the specifications in the ground segment specification tree
- Associate attributes of each requirement, such as
  - Owner (i.e., who is responsible for the requirement)
  - Type of requirement (e.g., functional, performance, specialty engineering, margins)
  - Specification section to which the requirement belongs
  - Whether the requirement is derived or not
  - Reference to source of the requirement if it is derived
  - Rationale for the requirement, including reference to documented analysis supporting that requirement, especially for quantitative requirements and their allocations to child requirements
- Associate each requirement with the verification method(s) specified for that requirement and the verification level(s) for each method, if applicable (i.e., level in the integration of the ground segment where the requirement is to be verified by the method)
- Provide the ability to maintain the verification status of each requirement throughout the test program (i.e., whether the requirement has been verified by each applicable verification method and in each applicable verification level)

- Provide the ability to link parent requirements and child requirements so that bi-directional traceability is maintained
- Provide a history of changes to requirements with associated information about the configuration control board (CCB) package that approved the change
- Maintain a log of changes for all areas of the database, including the development area as well as the current and archived baselines
- Provide the capability to roll back the changes to the previous entries
- Provide capabilities for the user to search and query the requirements and associated information
- Produce reports on errors, inconsistencies, omissions, and other problems across and within requirements, associated information, and traceability linkages
- Produce specifications from the requirements in the database for delivery as CDRL items
- Export requirements and associated data into tools such as Microsoft Word™ and Excel™
- Provide strong access control to the requirements database, including limiting functionality based on type of user
- Maintain environments for both classified and unclassified requirements, where applicable
- Produce metrics reports from the requirements database for use by managers, such as
    - Number of TBDs, TBRs, and TBSs remaining in the requirements for each specification
    - Number of requirements in each specification over time
    - Number and type of problems found over time
    - Number of requirements added, deleted, and modified over time for each specification

Establishing and Maintaining Requirements

The activity of establishing and maintaining requirements receives approved initial requirements and requirements changes from the configuration management process. This activity is responsible for populating the initial approved versions of the specifications (establishing the current baselined versions) and for maintaining the approved versions of the specifications as changes are made (maintaining the current baselined versions).

As shown in Figure 16-9, the requirements management process must be strongly connected to the configuration management process so that only requirements approved by the appropriate CCB are placed in the current baselined versions of the specifications. In some cases, there may be an engineering review board (ERB) responsible for technical review of the

requirements before they are presented to the CCB, and in large ground segment development projects there may be multiple layers of CCBs.

Archiving Requirements

The activity of archiving requirements stores the current baselined versions of the specifications in an archive area of the database before the current baselined versions of the specifications are changed. This activity organizes and maintains the area of the database containing the archived versions of the specifications so that team members can easily retrieve past versions of individual requirements or entire specifications. This activity also usually includes maintaining an index of available archived versions of each specification.

Disseminating Requirements Information

The activity of disseminating requirements information provides notification about the requirements and their associated information to the entire requirements development and management team and ideally to both contractor and acquirer personnel. The requirements management team notifies the appropriate personnel of changes to the requirements and associated information, updates to the current baselines of requirements, and other information. In particular, this activity extracts metrics information from the requirements database, produces metrics reports, and distributes those reports to appropriate personnel.

Providing Support for the Requirements Management System

As with all complex tools, the RMS requires continual support by specialists on the requirements management team. Requirements management systems require both tool support by tool experts and requirements database structure support by requirements database experts.

The activity of providing tool support for the RMS includes the following:

- Determining the capabilities that the RMS must have
- Performing an evaluation of available requirements management systems
- Selecting and purchasing the RMS software and any necessary hardware
- Obtaining and maintaining a sufficient number of licenses for both contractor and acquirer personnel who need to use the tool
- Installing, configuring, and testing the RMS tool
- Maintaining the tool in an up-to-date state by obtaining, installing, configuring, and testing all updates to the tool software released by the vendor
- Obtaining sufficient training in the tool for the requirements development and management teams, and obtaining necessary training for the teams in new tool versions as they are released

- Troubleshooting tool problems and obtaining vendor support when needed
- Backing up database contents, database structure, and tool configuration data

The activity of providing requirements database structure support includes the following:

- Eliciting information from the requirements development and management teams about what information needs to be included in the database and how the database will be used to support the requirements development and other ground segment development activities
- Designing, implementing, and testing the structure of the requirements database, including all requirements attributes and other associated information
- Maintaining the database structure and updating it as needs change

16.4.1.3 Requirements Traceability

The identification and documentation of the derivation path (upward) and the allocation/flow down path (downward) of requirements in the requirements hierarchy. Requirements traceability is also performed between requirements and architectural elements, design elements, implementation elements, and test plans and procedures. (Adapted from clause 3.2520 of [5].)Maintaining bi-directional traceability between parent and child requirements is one of the most important functions of the RMS, second only to maintaining the requirements themselves. Bi-directional traceability is needed to ensure that all parent requirements are properly covered by the child requirements, that is, the ground segment entities at the lower level will, when implemented, meet the requirements at the higher level. It is also needed to ensure that all child requirements have a source, either one or more parents requirements or an applicable reference for derived requirements.

Another very important function of the bi-directional traceability is to assist in assessing the impact of requirements changes as they occur. Traceability information provides the capability to understand the extent to which changes to higher-level requirements, including TRD requirements, affect the ground segment subsystems, components, and hardware and software items. Thus, the traceability information assists contractor and acquirer management in estimating the cost and schedule impact of such changes and in keeping ground segment development plans up-to-date as changes occur.

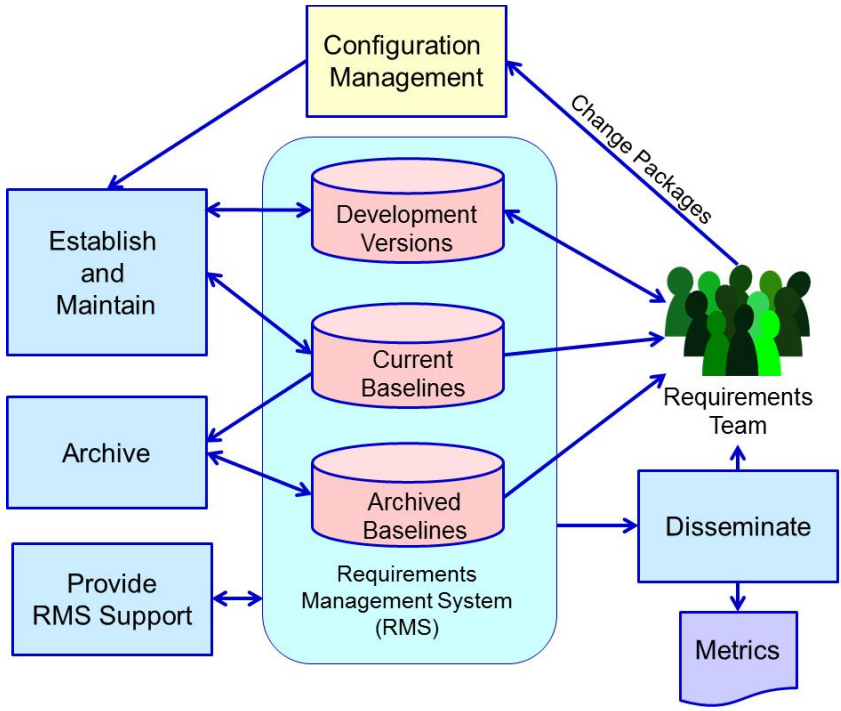


Figure 16-9. Ground segment requirements management process.

16.4.2 Ground Segment Requirements Engineering Lessons Learned

The following lessons learned apply to ground segment requirements engineering:

- The acquirer needs a requirements management system, as does the contractor, for use in developing the TRD requirements. Ideally, the acquirer and contractor should use the same RMS so that the TRD and requirements specifications may be easily transferred between the acquirer and contractor RMS environments.
- Requirements for the contractor to use an RMS must be included in the RFP to ensure the offerors bid the use of a central requirements database and a requirements management system. Usually such requirements are included in a clause under Section H, Special Provisions. If the acquirer wishes the contractor to use the same RMS as is implemented in the acquirer's organization, this must also be specified in this contract clause. See Abelson et al. work [18] for more information about the RMS Section H clause.

- In addition, the contract clause requiring the RMS should specify that the acquirer must have read access to current data in the contractor's RMS with no delay in obtaining up-to-date data, that is, the acquirer must have read access to the contractor's central requirements database, not some separate copy of the requirements. This is necessary to avoid delays in reviewing the requirements by the acquisition team.
- For large ground segment development efforts, the contractor team may consist of a prime contractor and a number of other team members that may be subcontractors, other divisions of the prime contractor and subcontractors, and possibly vendors performing development tasks. For teams with multiple team members, the same RMS should be used for the entire team, and personnel from all team members should be able to access the same current requirements database.

16.5 References

1. Adams, R. J., et al. *Software Development Standard for Mission Critical Systems*. TR-RS-2015-00012, The Aerospace Corporation, El Segundo, CA. March 17, 2014.

2. CMMI Product Team, *CMMI® for Development, Version 1.3 (CMMI-DEV, V1.3)*, Carnegie Mellon University (CMU)/Software Engineering Institute (SEI), CMU/SEI-2010-TR-033, November 2010.

3. Chairman of the Joint Chiefs of Staff (CJCS). Joint Capabilities Integration and Development System (JCIDS). CJCS Instruction (CJCSI) 3170.01, 23 January 2015.

4. International Organization for Standardization/International Electrotechnical Commission /Institute for Electrical and Electronic Engineers (ISO/IEC/IEEE). *Systems and Software Engineering—Vocabulary*. ISO/IEC/IEEE Standard 24765:2010, 15 December 2010.

5. International Council on Systems Engineering (INCOSE), *Systems Engineering Handbook Version 3.2.2.* INCOSE-TP-2003-002-03.2.2, October 2011.

6. ISO/IEC/IEEE, *Systems and Software Engineering – Lifecycle Processes – Requirements Engineering.* ISO/IEC/IEEE 29128, 1 December 2011.

7. Eslinger, Suellen, et al. *Integrating Software Assurance into the Request for Proposal*. TOR-2013-00694, The Aerospace Corporation, El Segundo, CA. January 31, 2014.

8. Guarro, S. et al. *Mission Assurance Guide (MAG)*. TOR-2007(8546)-6018 Rev. B, The Aerospace Corporation, El Segundo, CA. June 1, 2012.

9. DOD, *Operation of the Defense Acquisition System*. Department of Defense Instruction DODI 5000.02, 7 January 2015.

10. DoD, System/Subsystem Specification (SSS) Data Item Description. DI-IPSC-81431A, 10 January 2000. Revalidated 8 July 2013 (Notice 1).

11. DOD, Software Requirements Specification (SRS) Data Item Description. DI-IPSC-81433A, 10 January 2000. Revalidated 8 July 2013 (Notice 1).

12. DOD, Interface Requirements Specification (IRS) Data Item Description, DI-IPSC-81434A, 10 January 2000. Revalidated 8 July 2013 (Notice 1).

13. Shaw, B. E. *Systems Engineering Requirements and Products (A Revision of TOR-2005(8583)-3 REV B).* TR-2013-00001, The Aerospace Corporation, El Segundo, CA. February 28, 2013.

14. Institute for Electrical and Electronic Engineers (IEEE), *Draft Standard for Application of Systems Engineering on Defense Programs.* IEEE P15288.1/D4.1, September 2014.

15. Peresztegy, L. B. and C. E. O'Connor. *Technical Reviews and Audits of Systems, Equipment, and Computer Software*. TOR-2007(8583)-6414, Rev. 1, Vol. 1, The Aerospace Corporation, El Segundo, CA. January 30, 2009.

16. Peresztegy, L. B. and C. E. O'Connor. *Technical Reviews and Audits of Systems, Equipment, and Computer Software.* TOR-2007(8583)-6414, Rev. 1, Vol. 2, The Aerospace Corporation, El Segundo, CA. January 30, 2009.

17. IEEE, *Draft Standard for Technical Reviews and Audits on Defense Programs.* IEEE P15288.2/D5.2, September 2014.

18. Abelson, Linda, A., et al. *Integrating Software Topics into the Request for Proposal.* TOR-2011(8506)-117, The Aerospace Corporation, El Segundo, CA. July 19, 2012.

19. Owens, Karen L., and J. M. Tagami. *Recommended Software-Related Contract Deliverables for National Security Space System Programs.* TOR-2006(8506)-5738, The Aerospace Corporation, El Segundo, CA. February 14, 2008.

20. Owens, Karen L., and J. M. Tagami. *Recommended Software-Related Systems Engineering Contract Deliverables for National Security Space System Programs.* TOR-2005(8506)-8101, The Aerospace Corporation, El Segundo, CA. June 27, 2008.

16.6 Acronyms

A	analysis (verification method)
ATP	authority to proceed
B	Acquisition Milestone B
C&C	command and control
CCB	configuration control board
CDD	capability development document
CDR	critical design review
CDRL	contract data requirements list
CJCS	Chairman of the Joint Chiefs of Staff
CJCSI	chairman of the joint chiefs of staff instruction
CMMI®	capability maturity model® integrationsm
CMU	Carnegie Mellon University
CONOPS	concept of operations
COTS	commercial off-the-shelf
CSCI	computer software configuration item
D	demonstration (verification method)
D&R	detection and resolution
DID	data item description
DOD	Department Of Defense
DODI	Department of Defense Instruction
EMD	engineering and manufacturing development
ERB	Engineering Review Board
FD	full deployment
FOC	final operation capability
FFRDC	federally funded research and development center
Gov't	government
Grnd	ground
HRS	hardware requirements specification
HW	hardware
I	inspection (verification method)
ICD	initial capabilities document
IEC	International Electrotechnical Commission
IEEE	Institute for Electrical and Electronic Engineers
INCOSE	International Council on Systems Engineering
IOC	initial operation capabilitiy
IRS	interface requirements specification
ISO	International Organization for Standardization
JCIDS	joint capabilities integration and development system

MAG	Mission Assurance Guide
MDD	materiel development decision
MSA	materiel solution analysis
O&S	operations and support
OT&E	operational test and evaluation
PDR	preliminary design review
Regs	regulations
Reqs	requirements
RFP	request for proposal
RMA	reliability, maintainability, and availability
RMS	requirements management system
S	special (verification method)
SAR	software requirements and architecture review
SDR	system design review
SE&I	systems engineering and integration
Seg	segment
SEI	Software Engineering Institute
SETA	systems engineering and technical assistance
SFR	system functional review
SOW	statement of work
Spec	specification
Specs	specifications
SRR	system requirements review
SRS	software requirements specification
SSS	system/subsystem specification
SW	software
T	test (verification method)
TBD	to be determined
TBR	to be resolved or to be revised
TBS	to be specified or to be supplied
TMRR	technology maturation and risk reduction
TOR	technical operating report
TP	telemetry processing
TR	technical report
TRD	technical requirements document
TRR	test readiness review

Chapter 17
Software Architecture

Alan D. Unell
Computers and Software Division
Engineering and Technology Group
Geraldine A. Chaudhri
Software Systems Assurance Department
Computer Applications and Assurance Subdivision

17.1 Introduction

The program office team supports the development of the ground segment software architecture, including conducting an independent architecture evaluation as well as establishing best practices for ensuring that the software architecture meets it intended needs.

17.2 Definitions

Software architecture A blueprint for a system's software.

Software architecture evaluation A review of the software architecture to assess the risks and strengths usually before development has begun.

Model driven engineering A software development methodology which focuses on creating and exploiting domain models, which are conceptual models of all the topics related to a specific problem.

17.3 Software Architecture Tasks and Principles

17.3.1 What is Software Architecture and Why Evaluate it?

Software architecture is a simplified representation, or a model, of a software system. It is also described as "the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them." [1]

Software architecture is important to large, complex, software-intensive systems because it is a:

- Vehicle for communication among stakeholders. The architecture model can be used as a basis for creating mutual understanding, forming consensus, and communicating with each other.

- Manifestation of the earliest design decisions. Software architecture is the earliest artifact that enables the priorities among competing concerns to be analyzed.

- Reusable, transferable abstraction of a system. The software architecture can be applied to other systems exhibiting similar requirements and can promote large-scale reuse and software product lines [2].

A key challenge in modeling is to identify what aspects of the system to capture; this is generally driven by stakeholder and domain-specific concerns. A typical software system may require several simultaneous national security space (NSS)-specific attributes, such as commanding a vehicle and processing sensor data while being resilient to attacks, scalable to meet peak and future usage, flexible to incorporate new capabilities, and timely and reliable to support the war fighter. The key stakeholders may include DOD enterprise representatives, contractors, operators, commercial users, product vendors, and subject-matter experts.

In addition to providing the basis for the system's technical performance, the software architecture provides for "-ilities", such as reliability, maintainability, reuse and other important characteristics. The architectural design principles, requirements, constraints, and assumptions serve as formal guidance to development engineers during the preliminary and detailed design phases of the software development lifecycle. It is not uncommon, during these phases, to discover initial design constraints and assumptions that must be revised in order to provide detailed designs and implementations that can meet target system requirements. Such cases can occur due to changes to existing requirements or perhaps the addition of new requirements.

Often revisions to architectural designs during lower-level design phases are quickly performed in order to remove development roadblocks, but potentially without full regard or understanding of the impacts to other areas of the architecture. Conducting broad or even targeted architectural evaluations at regular intervals during the lower-level design and implementation phases of software development can provide greater assurance to stakeholders that the evolving architectural design continues to meet all functional and non-functional system requirements.

17.3.2 Building a Software Architecture

Typical development has a systems engineering team building an architecture which is then elaborated into a software architecture. The lead software engineers should be active participants with systems engineers in the

architecture development so that they do not preclude valuable software solutions. The following are key steps in developing a software architecture:

- Identify the requirements and/or needs that are key to solving the problem. This includes not only how fast or precise or robust the system must be, but also the connectivity to other systems, and the quality and performance attributes that are desired.

- Communicate with stakeholders (i.e., customers, users). This will help bring to light or elaborate any hidden desires or concepts of operations that need to be considered when solving the problem.

- Identify the major components and functions that need to be performed by the system. Major components and functions will track how the different candidates for each of these major blocks of the system could be put together in order to determine a particular solution for a building block, or for the system as a feasible model. Such a model may be a discrete event simulation (checks for speeds and resource usage), a coarse spreadsheet (checks basic timing), or a dynamic logic (checks the type of model that may give a probabilistic view of the architecture's chance of success).

- Identify risks. As the team develops the candidate software architecture solutions any risks or constraints that may arise should be tracked as well as any trade studies performed. These enablers and/or limiters will be important to recall in the future.

17.3.3 Communicating the Software Architecture

After the key architectural decisions are made and the architectural development steps are complete, it is important to document the software architecture using either a model-driven engineering tool or a suite of desktop tools. Regardless of which tools are used, the architecture should be depicted in a way that addresses the needs of all the stakeholders. This may be accomplished through a set of views. The 4+1 is a view model designed by Philippe Krutchen for "describing the architecture of software-intensive systems, based on the use of multiple, concurrent views [3]." Figure 17-1 depicts this model. The views are used to describe the system from the viewpoint of different stakeholders, such as end-users, developers, and project managers. The four views of the model are logical, development, process, and physical view. In addition, selected-use cases or scenarios are used to illustrate the architecture serving as the 'plus one' view. Hence the model contains 4+1 views.

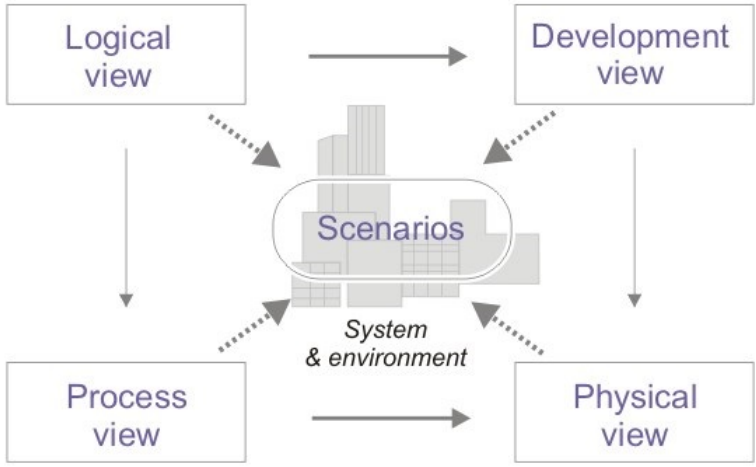


Figure 17-1. The 4+1 view model [4].

***Logical view*:** The logical view is concerned with the functionality that the system provides to end-users. Unified modeling language (UML) diagrams used to represent the logical view include class diagrams, communication diagrams, and sequence diagrams

***Development view*:** The development view illustrates a system from a programmer's perspective and is concerned with software management. This view is also known as the implementation view. It uses the UML component diagram to describe system components and the UML package diagram to represent the overall development view.

***Process view*:** The process view deals with the dynamic aspects of the system, explains the system processes and how they communicate, and focuses on the runtime behavior of the system. The process view addresses concurrency, distribution, integrators, performance, and scalability, etc. UML diagrams to represent process view include the activity diagram.

***Physical view*:** The physical view depicts the system from a system engineer's point of view. It is concerned with the topology of software components on the physical layer, as well as the physical connections between these components. This view is also known as the deployment view. UML diagrams used to represent physical view include the deployment diagram.

***Scenarios*:** The description of an architecture is illustrated using a small set of use cases or scenarios which become a fifth view. The scenarios describe sequences of interactions between objects, and between processes. They are

used to identify architectural elements and to illustrate and validate the architecture design. They also serve as a starting point for tests of an architecture prototype. This view is also known as the use case view.

Another resource for guidance on documenting software architecture is IEEE-STD-1471- 2000 (superseded by ISO/IEC/IEEE 42010:2011) "Practice for Architectural Description of Software-Intensive Systems." [5, 6] This standard was developed by the IEEE architecture working group with representation from industry, academia, and other standards bodies. It provides a conceptual framework for architectural description and is designed to be independent of any individual architectural description technique.

During design and evaluation a common language is established in which different program concerns can be expressed, negotiated, and resolved: for example, creating program-tailored Department of Defense Architecture Framework (DODAF), and UML instantiations following IEEE-42010. Without such a language, it becomes difficult to share the program design philosophy among the stakeholders and development team to guide the day-to-day development decisions that ultimately influence both quality and usefulness of the end product.

With a well-established communication medium, software architecture evaluations can occur regularly to ensure that evolving program and stakeholder needs continue to be satisfied. Such reviews can provide ongoing insight into the potential impacts of new or changed requirements, or design constraints.

17.3.4 Evaluating the Software Architecture

There are a number of industry methods for evaluating software architecture. One widely adopted method is the Architectural Tradeoff Analysis Method (ATAM)®, developed by the Software Engineering Institute. The purpose of this method is to assess the consequences of architectural decisions in light of quality attribute requirements and business goals. The method provides a set of steps to help stakeholders ask questions to discover potentially problematic architectural areas, using scenario-based assessments early in a program to address quality attributes such as modifiability, performance, and availability. It is aimed at raising awareness of critical issues, localizing and analyzing tradeoffs, and focusing in areas of highest risk (to read about other methods see, "From Methods for Evaluating Software Architecture: A Survey", 2008, Banani Roy and T.C. Nicholas Graham [7]). ATAM® and similar methods focus primarily on the process of performing a software architecture evaluation and are not targeted to specific software applications.

The Aerospace Corporation's Software Architecture Evaluation Framework (hereafter, 'Framework') is different from ATAM, but complementary in that it

provides a concrete set of questions and evaluation guidance tailored specifically to NSS systems software. The Framework is composed of an extensive list of "telling" questions to ask about a software architecture. Questions are grouped into four, top-level categories: architecture fundamentals, architecture documentation, architectural functionality and quality attributes, and architecture development and evolution methodology. Each category is further broken down into dimensions. Dimensions represent areas of concern or evaluation criteria. Dimensions include conventional software quality attributes such as scalability and availability, as well as concerns specific to NSS programs such as, re-programmability, resilience to cyber attack, and commercial off the shelf (COTS)/government off the shelf (GOTS) appropriateness. There are at least 40 dimensions defined thus far in the framework.

In addition to having NSS-specific dimensions, the Framework questions in all dimensions are written to suit NSS systems. Questions are defined at three levels. At Level 1 (generic level), questions are non-domain specific and they can be applied to most software systems. These Level 1 questions provide a basis for discussions between subject matter experts (SMEs) and software experts to refine the generic questions to become Level 2 questions. Level 2 questions pertain to NSS domains (e.g., command and control or mission planning). The evaluation team of a given system can then tailor Level 1 and Level 2 questions based on the requirements of the system under evaluation, creating NSS system-specific (Level 3) questions.

Armed with the framework of questions that are well-tuned to the system domains, evaluators are enabled to support NSS systems in the following steps:

- Selection of evaluation criteria that are important to a given NSS system. The dimensions (or the evaluation criteria) are carefully selected with the characteristics of NSS systems in mind.

- Interpretation of the evaluation criteria for various NSS system domains. The generic evaluation questions are further tailored to make them NSS-system-specific. For example, scalability of a ground system might be interpreted as the capability to support a larger number of space vehicles. This is quite different from the scalability of an information technology (IT) system.

- Improved program evaluations of current and next generation software systems. Improvements have been made through the many decades of SME experiences in building NSS systems that are combined with software expertise that is built into the framework. The framework

developer team consists of several architects and/or engineers with decades of experiences in building or oversight of NSS systems.

The dimensions and questions selected by the evaluation team should help to determine the quality and effectiveness of the software architecture. Some characteristics of an effective software architecture include the following:

- Modules are loosely coupled so that updates to the COTS products do not impact the interfaces
- Software is portable and platform independent (i.e., from hardware and operating system)
- Software has proven reliability and availability attributes—incorporates redundancy and is fault tolerant which reduces operations cost
- Software is extensible and flexible to ensure minimum modification when adding or replacing COTS

The Framework can be used at different phases or milestones throughout a program's lifecycle. The manner in which the framework is applied, as well as the benefits achieved, will vary depending on the particular phase or milestone. As an example, during the early project inception and pre-systems acquisition phase, architectural design questions provided by the framework serve as guidance to help inform, as well as provide constraints on, the definition of system concepts that will ultimately serve as the foundation for the creation of system-level requirements.

The Framework provides capabilities that enable full evaluation of complete and existing operational systems. It can also be tailored to address the specific level of architectural design detail commonly expected to be available at a particular review milestone such as system design review (SDR), preliminary design review (PDR), and critical design review (CDR). For example, a software architecture evaluation using the framework in support of a PDR milestone will prove invaluable in providing technical reviewers with a detailed picture of the contractor's architectural design, as well as providing supplemental input into the formal PDR review process.

Since its inception in 2010, the Framework has been successfully used by several NSS programs. The software architecture of a ground system, currently under development, was evaluated using the framework as part of the PDR campaign. This evaluation utilized the risks and/or opportunities that were identified by an earlier ATAM-based evaluation on the program. During the evaluation scoping phase, the ATAM-identified risks were used in selecting the evaluation criteria. This experience illustrated the complementary nature of the Framework and scenario-based evaluation methodologies such as ATAM. In another program evaluation, the performance issues facing an NSS system were

analyzed. This methodic analysis of the system performance was instrumental in timely identification of the sources of the performance issues. This could not have been possible using any ad-hoc software design analysis. In yet another different setting, the Framework was used to identify a set of potential software architecture/design areas that needed to be evaluated during the source selection process. These examples helped the evaluation development teams to: (i) validate the utility of the framework for various software architecture/design evaluation purposes; and (ii) enhance the framework with the lessons learned.

17.4 Practices

The program office team must plan for software architecture evaluation activities early on in the acquisition lifecycle. The request for proposal (RFP) should contain language to specify that a software architecture is to be produced and it should include a data item description (DID) describing its content.

The system and software architectures are evaluated when proposals are submitted. Recommended specific language for RFP sections is contained in section 3.7 of TOR-2011(8506)-117, "Integrating Software Topics into the Request for Proposal", July 19, 2012 [7]. Special Contract Requirements for Software Architecture Evaluations are specifically detailed. It is recommended that section H in the RFP address the following:

- A description of the evaluation method to be used and its steps
- A description of the participants in the architecture evaluation and their roles and responsibilities
- A specification of the number and frequency of evaluations to be conducted
- If multiple evaluations are involved, a description of how are they will be staged
- Entrance and exit criteria for conducting the evaluations
- A description of what is involved in terms of time, effort, and cost
- An explanation of how the objectivity of the participants will be ensured
- A description of how the evaluation results will be captured and used
- A description of what deliverables need to be included in the evaluation
- A description of how the evaluations will be carried out collaboratively to ensure both government and contractor stakeholders play an active role
- A description of the training to be provided to evaluation team members

In addition, the contractors are asked to provide a software architecture description (SAD) document along with their proposals. During the source

selection, the acquisition team will perform a preliminary evaluation of the software architectures presented in these SADs. It is also recommended that software architecture evaluations, such as the ATAM®, and support to government led evaluations are included as contract line items to the contract.

After contract award, the acquisition team works with the contractor to perform the ATAM®. The government team then begins plans to conduct an independent evaluation using the Framework described above. These evaluations would then be performed at key project milestones, such as Software Requirements Review (SRR), PDR, and CDR. The acquisition team keeps careful records of the resulting actions and recommendations from the evaluations and identifies all risks. Action items are then tracked to completion and risks mitigated before each successive evaluation. As the project enters its development stage, the acquisition team must make frequent checks to ensure that the architecture model is updated and maintained throughout the development cycle. In fact, this same practice must be applied throughout the project's entire lifecycle.

17.5 Key Lessons Learned

Next to having a solid set of requirements, developing an effective software architecture is critical to the successful development of ground segment software components. The decisions that are made when developing the architecture have far-reaching and long-lasting effects that impact the reliability, sustainability, security, and overall performance of the ground segment components.

Some key lessons that have been learned about conducting software architecture evaluations are following:

- Select the most important driving dimensions (probably no more than 10) and carefully tailor telling questions about the program related to those dimensions
  - Use the program's contract and set of requirements to guide the selection
- Include customers and other key stakeholders in the development of the architecture. Such stake holders also include software, users, maintainers, and operators. Their inputs can help guide the solution space.
- Have a clear set of architecture attributes that drive the architecture to evaluate
- Include customer team on the evaluation
- Evaluate the architecture often and especially at key milestones, system readiness review (SRR), PDR and CDR

- Use the same architecture modeling tool suite that the contractor is using to avoid communication issues and to keep track of updates
- Track and follow up with all actions, recommendations and risks that result from the evaluations

17.6 Government and Contractor Enabling Processes and Products

Architecture development should begin early on in the acquisition phase. During the RFP phase, a SAD document is submitted by each contractor and is evaluated by the source selection team. Once on contract, the government team and the contractor work together deciding on the most critical quality attributes in the architecture. The contractor will then develop a software architecture and catalogue the trade studies they've done to arrive at their architecture. The results of this initial architecture development are input to the Software Architecture Evaluation Framework study which is conducted solely by the government team. The government team also uses the project requirements, architecture description, and the contract itself to define the breadth and scope of the evaluation. The government must be sure to allocate budget and schedule for the contractor to support this activity. In addition, the contractor must provide access to all of the architecture products generated by their engineering modeling tools to the evaluation team. In some cases, the government team may have access to the architecture models directly through a contractor portal. A software architecture evaluation is performed at key milestones in the project leading up to the start of software code and unit test. All actions and risks are documented and tracked through closure. Table 17-1 summarizes these activities and their resulting products.

Table 17-1. Software Architecture Evaluation Activities and Products

Activity	When Performed	Products
Contractor delivers Software Architecture Description Document	In response to RFP, at SRR, PDR, CDR, final version at system acceptance	Software Architecture Description Document
Government evaluates initial software architecture	Source selection	Technical rating
Government and contractor identify critical architectural features	Prior to SRR	Out brief, list of actions

Activity	When Performed	Products
Government conducts architecture evaluation	Prior to SRR, PDR, and CDR	Detailed responses to questions, evaluation results in a technical report and formal briefing
Contractor is briefed on results of evaluation	At completion of evaluations after SRR, PDR, and CDR	Action items, risks, recommendations
Action items and risks tracked through closure	Throughout architecture development period up to start of code and unit test	Evidence of action item resolution and risk mitigation activities

17.7 References

1. Bass, L., P. Clements, R. Kazman, *Software Architecture in Practice*. Reading, MA: Addison-Wesley, 1998.

2. Clements, P., R. Kazman, Mark Klein, *Evaluating Software Architectures, Methods and Case Studies*, Boston, MA: Addison-Wesley, 2011.

3. Kruchten, Phillipe, "Architectural Blueprints—The "4+1" View Model of Software Architecture," IEEE Software 12 (6) November 1995, pp. 45-50.

4. "4+1 Architectural View Model" by Marcel Douwe Dekker - Own work. Licensed under CC BY-SA 3.0 via Creative Commons, https://commons.wikimedia.org/wiki/File:4%2B1_Architectural_View_Model.jpg

5. ANSI/IEEE 1471-2000, Recommended Practice for Architecture Description of Software-Intensive Systems.

6. ISO/IEEE/IEC 42010:2011, Systems and Software Engineering—Architecture Description.

7. Roy, Banani, and T.C. Nicholas Graham, *Methods for Evaluating Software Architecture: A Survey*, Ontario, Canada, Queen's University at Kingston Technical Report No. 2008-545, 2008.

17.8 Bibliography

Clements, P., F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord, and J. Stafford, "Documenting Software Architectures: Views and Beyond."

Garland, P.J., R. Anthony, “Large-Scale Software Architecture, A Practical Guide using UML”

Department of Defense Architecture Framework, Version 2.02, Volumes I, II, and III.

Taylor, R. N., N. Medvidovic, and E. M. Dashofy, Software Architecture: Foundations, Theory, and Practice.

Unell, Alan. “Evaluating Software Architectures for National Security Space Systems”, The Aerospace Corporation, El Segundo, CA. *Crosslink Magazine*, Spring 2013

Meyers, Steven A. and Alan Unell. “Software Architecture Reviews Improve Development.” The Aerospace Corporation, El Segundo, CA. Getting It Right, Vol 3 Issue 2, Nov 30, 2012,

Software Engineering Institute, Architecture Trade Off Analysis Method (ATAM), http://www.sei.cmu.edu/architecture/tools/evaluate/atam.cfm.

Abelson, Linda A. et al., Integrating Software Topics into the Request for Proposal, The Aerospace Corporation, El Segundo, CA, TOR-2011(8506)-117, 2012.

17.9 Acronyms

ATAM	architectural tradeoff analysis model
CDR	critical design review
COTS	commercial-off-the-shelf
DID	data item description
DOD	Department of Defense
DODAF	Department of Defense Architecture Framework
GOTS	government off-the-shelf
IT	information technology
NSS	National Security Space
PDR	preliminary design review
RFP	request for proposal
SAD	software architecture description
SDR	system design review
SME	subject matter expert
SRR	software readiness review
UML	Unified modeling language

Chapter 18
Product Development

Suellen Eslinger
Software Engineering Subdivision
Computers and Software Division

18.1 Introduction/Background

This chapter addresses the development, integration, and test of the various products that comprise a ground segment. In this handbook, the ground segment consists of a set of subsystems, each of which consists of a set of components, hardware items, and/or software items. The components may themselves consist of hardware and software items, or they may be single entities (e.g., a commercial off-the-shelf [COTS] front-end communication protocol handling system). Figure 18-1 shows the overall ground segment architectures. The hierarchical decomposition of an example ground segment element is shown in Figure 18-2.

Figure 18-1 shows the ground segment as composed of a number of subsystems: a space/ground asset command and control subsystem, a mission management subsystem, a mission data processing and distribution subsystem, an infrastructure services subsystem, and perhaps other subsystems. The space/ground asset command and control subsystem is shown divided into components: a telemetry processing component, a space vehicle command and control component, a ground asset command and control component, and perhaps other components. The subsystem may also include hardware item(s) and/or software item(s) that are not contained within the parent components. Each component may also be divided into hardware item(s) and/or software item(s).

This chapter describes the development of hardware and software items and their integration into components and subsystems. Integration of the subsystems into the complete ground segment and the end-to-end testing of the integrated subsystems are addressed herein.

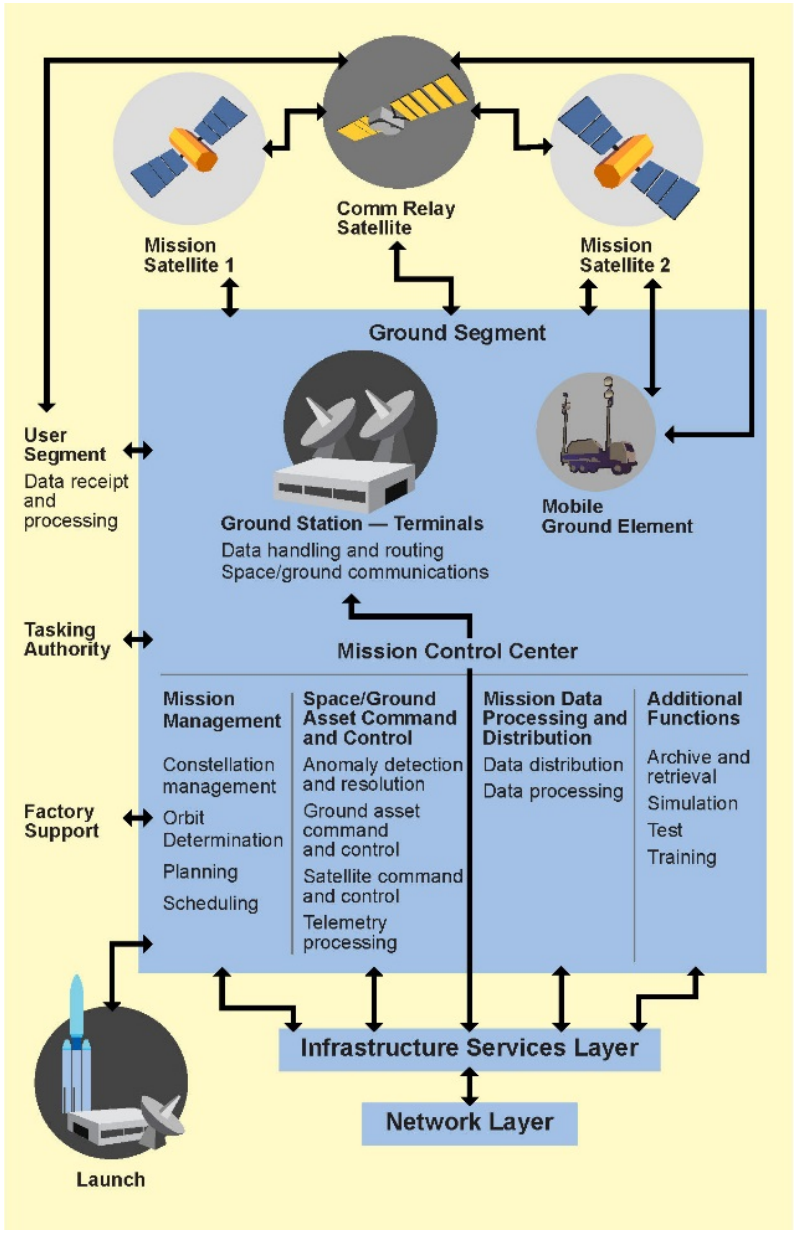


Figure 18-1. Ground systems architecture.

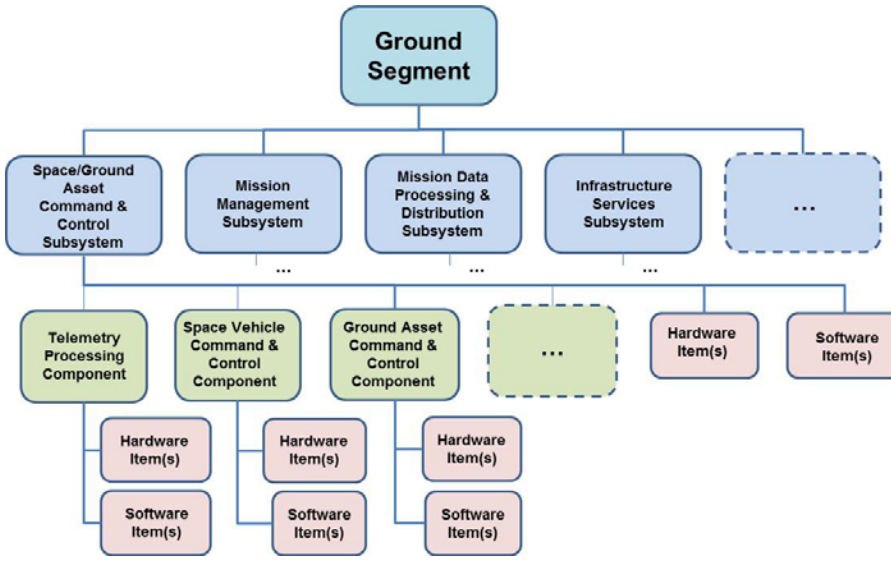


Figure 18-2. Example ground segment decomposition.

18.2 Definitions

Acquirer An organization that procures products for itself or another organization [1].

Acquisition team The acquisition team includes the acquirer personnel as well as Federally Funded Research and Development Center (FFRDC), Systems Engineering and Technical Assistance (SETA), and Systems Engineering and Integration (SE&I) contractor personnel.

Agile development lifecycle model A group of software development methods based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. The model promotes adaptive planning, evolutionary development and delivery, and is a time-boxed iterative approach, encouraging rapid and flexible response to change. It is a conceptual framework that promotes foreseen interactions throughout the development cycle. (Adapted from Wikipedia [2].)

Assembly A portion of a hardware unit that can be provisioned and replaced as an entity and which normally incorporates replaceable parts or groups of parts.

Baseline A product or a set of products that has been formally reviewed and agreed on at a particular point in the item's lifecycle, which thereafter serves as

the basis for further development, and which can be changed only through change control procedures. (Adapted from [3].)

Build A version of the system or software that meets a specified subset of the requirements that the completed system or software will meet [1]. Contractors use various synonyms for the term "build", such as "increment", "release", "block", "iteration", "cycle", "drop", "spiral", and "sprint". In this chapter, the term "build" will be used for software development iterations, and the term "increment" will be used for the ground segment development iterations that may include hardware, software, and facilities.

CDRL item The term is used for the list of deliverable data in a contract. Data here refers to information, usually described in documents. An individual deliverable data product is called a contract data requirements list (CDRL) item. The CDRL does not include the operational hardware and software to be delivered for the ground segment and installed in the operational facility.

Component An entity in the architecture of a ground segment subsystem. Components may consist of hardware and/or software items, or may be procured as a single entity.

Developer An organization that develops products ("develops" includes new development, modification, integration, reuse, reengineering, maintenance, or any other activity that results in products) for the contract. The term "developer" encompasses all contractor team members [1].

Discrepancy Any condition that deviates from expectations based on requirements specifications, architecture documents, design documents, user documents, plans, procedures, reports, standards, policy, etc., or from a user's or other stakeholder's sound engineering judgment or experiences. Discrepancies can be found during, but not limited to, the review, test, analysis, compilation, or use of products or applicable documentation. The term discrepancy is used throughout this chapter where others might use terms such as anomaly, defect, error, fault, failure, incident, flaw, problem, gripe, glitch, or bug. (Adapted from Adams et al. [1].)

Discrepancy Report Documentation of discrepancies and test incidents. Discrepancy reports are written for any potential discrepancy or test incident, even if the final resolution is that no discrepancy exists. Discrepancy reports are sometimes called problem reports, trouble reports, test incident reports, issue reports, and other terms. (Adapted from Adams et al. [1].)

Evolutionary software development lifecycle model A software development lifecycle model in which requirements development is done in each build. Each build also contains architecture, detailed design, implementation and unit

testing, integration and testing, and qualification testing in an overlapping, iterative manner, resulting in evolutionary requirements and incremental completion of the overall software product. Each build is not necessarily delivered to the acquirer. (Adapted from Adams et al. [1].)

Firmware The combination of a hardware device with computer instructions, computer data, or both that reside as read-only software on the hardware device [1].

Hardware item An aggregation of hardware that satisfies an end-use function and is designated for specification, interfacing, qualification testing, configuration management, or other purposes [1].

Hardware unit A portion of a hardware item that can be provisioned and replaced as an entity.

Incremental software development lifecycle model A software development lifecycle model in which all software requirements development occurs first, followed by a series of builds in which architecture, design, implementation and unit testing, integration and testing, and qualification testing occur in an overlapping, iterative manner, resulting in incremental completion of the overall software product. Each build is not necessarily delivered to the acquirer. (Adapted from Adams et al. [1].)

Integral activities Activities that are essential to the quality and completion of the products produced by the product-oriented activities, but that do not produce those products themselves. Integral activities are performed concurrently with the product-oriented activities throughout the development lifecycle. Examples of integral activities are configuration and data management, peer reviews, and other types of technical reviews, and risk management. (Adapted from Adams et al. [1].)

Iterative lifecycle model A lifecycle model, such as incremental or evolutionary with repeating and possibly overlapping development activities. Note: Iterative software development lifecycle models include agile development. (Adapted from Adams et al. [1].)

Lifecycle model A project management framework providing a sequencing strategy and a disciplined approach to the structure and order of activities

Process A set of interrelated activities, which transform inputs into outputs, to achieve a given purpose [3].

Product Information, software, or hardware created, modified, or incorporated by means of systems engineering, integration and test or software or hardware

development activities. Examples include plans, requirements, architecture, design, code, hardware units, databases, test information, and manuals. (Adapted from Adams et al. [1].)

Product-oriented activities Activities that directly produce an intermediate or final product of the ground segment

Quality enhancement activities Activities whose purpose is finding discrepancies in products so that the identified discrepancies can be fixed via the corrective action system. Examples of quality enhancement activities are peer reviews, technical reviews, and quality assurance reviews.

Software assurance The planned and systematic set of activities that ensures that software processes and products conform to requirements, standards, and procedures to help achieve:

- Trustworthiness - No exploitable vulnerabilities exist, either of malicious or unintentional origin, and
- Predictable Execution Justifiable confidence that software, when executed, functions as intended [4].

Software development A set of activities that results in software products. Software development includes new development, modification, reuse, reengineering, maintenance, and any other activities that result in software products [1].

Software development files A repository for material pertinent to the development of a particular body of software. Contents typically include (either directly or by reference) the development products themselves, as well as considerations, rationale, and constraints related to requirements development, architecture, design, and implementation; test information, including test cases and test results; discrepancy and change reports; technical reports; notes; and schedule and status information. Note: The body of software may be at different levels of integration (e.g., software unit, collection of related software units, software build, and software item). (Adapted from Adams et al. [1].)

Software engineering "The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software" [5].

Software item An aggregation of software that satisfies an end-use function and is designated for specification, interfacing, qualification testing, configuration management, and other purposes. Software items are selected based on tradeoffs among software function, size, host or target computer systems, developer,

support strategies, plans for reuse, criticality, interface considerations, the need to be separately documented and controlled, and other factors. A software item is composed of one or more software units. A software item is sometimes called a computer software configuration item (CSCI) [1].

Software unit An element in the design of a software item, for example, a major subdivision of a software item, a component of that subdivision, a class, object, module, function, routine, or database. Software units might occur at different levels of a hierarchy and might consist of other software units. Software units in the design might or might not have a one-to-one relationship with the code and data entities (e.g., routines, procedures, databases, data files) that implement them or with the computer files containing those entities [1].

Subassembly A portion of an assembly, consisting of two or more parts that can be provisioned and replaced as an entity.

Subcontractor Within this chapter, the term "subcontractor" is interpreted to mean any ground segment development team member tasked by the prime contractor or another team member to perform part of the required ground segment effort. This definition is broader than the usual legal definition of subcontractor. It includes, for example, other divisions of the prime contractor or other team members.

Subsystem An entity in the architecture of a ground segment. Subsystems generally consist of components, hardware items, and/or software items.

Stakeholder A group or individual that is affected by or is in some way accountable for the outcome of a product. (Adapted from [3].)

Waterfall lifecycle model A system or software lifecycle model in which the constituent activities are performed once in sequential order, possibly with overlap, but with little or no iteration. For development, the activities typically include requirements development, architecture, detailed design, implementation/fabrication/procurement and unit testing, integration and testing, and qualification/verification testing. (Adapted from Adams et al. [1].)

18.3 Activities in Ground Segment Product Development

There are core activities performed by the development team for the ground segment hardware and software products. These core activities are divided into two types: product-oriented and integral. The product-oriented activities (Figure 18-3) are those activities that directly produce an intermediate or final product of the ground segment. The integral activities (Figure 18-4) are activities that are performed concurrently with the product-oriented activities and are essential to their proper completion. The integral activities do not directly produce the

products; they do, however, directly affect the preparation, content, and quality of those products.

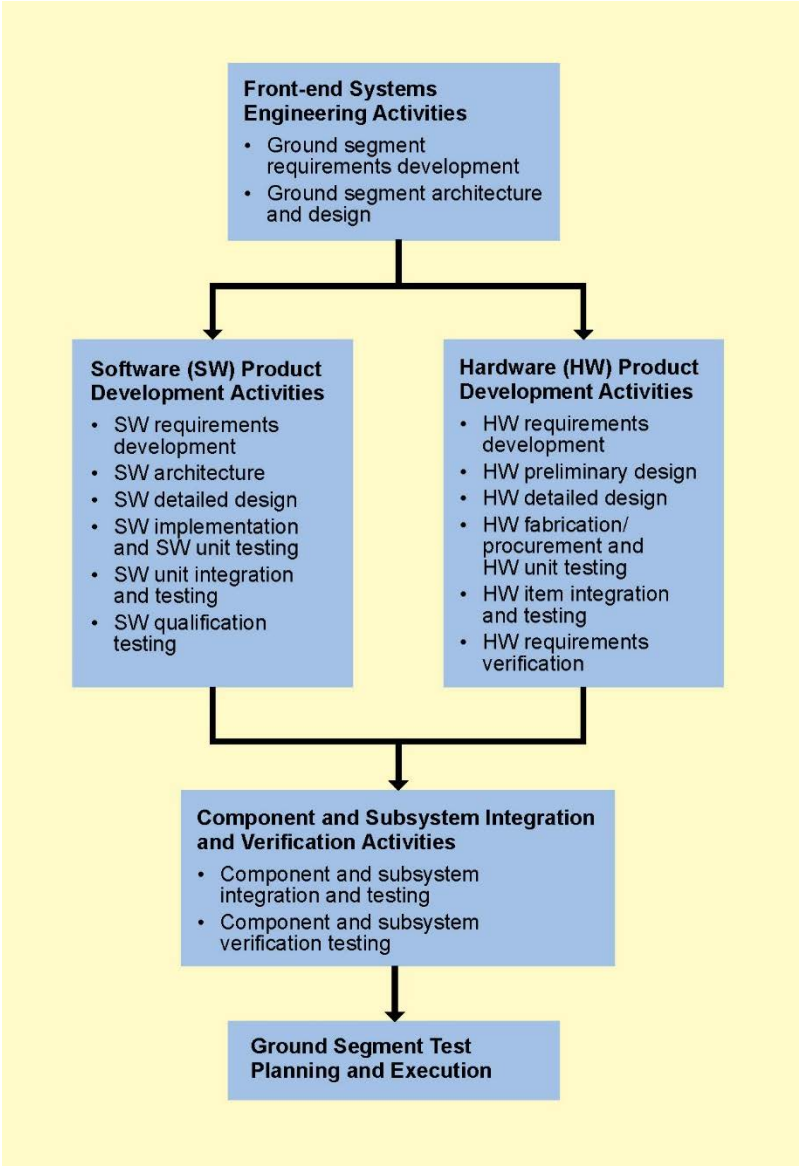


Figure 18-3. Ground segment development product-oriented activities.

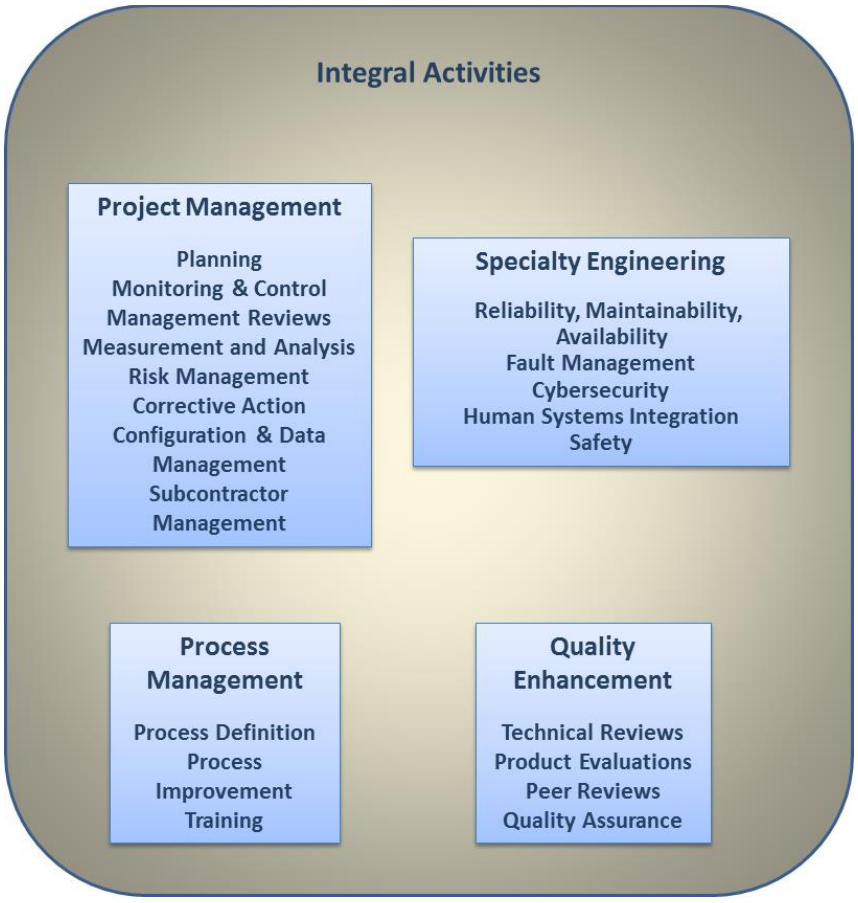


Figure 18-4. Ground segment development integral activities.

Note that Figure 18-4 is not intended to imply an order of execution or concurrence of the product-oriented activities, but simply to indicate what the major groups of activities are. The order of execution of the activities is determined by the development lifecycle model(s) in use.

18.3.1 Product-oriented Core Activities

The product-oriented core activities in ground segment development include front-end systems engineering, software product development, hardware product development, component and subsystem integration and verification, and ground segment test planning and execution activities (see Figure 18-3).

18.3.1.1 Software and Hardware Participation in Front-end Systems Engineering Activities

In addition to systems engineering personnel, development team and acquisition team personnel who are experts in the software and hardware disciplines that are involved in the ground segment development must participate in the front-end systems engineering activities. These front-end systems engineering activities consist of ground segment requirements development and ground segment architecture and design, which are performed at the segment, subsystem, and component levels, as applicable to the specific ground segment under development. Requirements development and architecture and design are performed iteratively and concurrently down through the specification tree to the level of the individual components and hardware and software items.

Table 18-1 briefly describes each front-end systems engineering activity and lists the usual products produced by the activity. The phrase "participate in" is used in the description of the activity because the activity is commonly the responsibility of systems engineers, and software and hardware development personnel take part in that activity.

It is important that the software and hardware acquisition team personnel review the products produced by these front-end systems engineering activities to ensure that (1) the technical requirements document (TRD) requirements are appropriately flowed down through the specification tree to the individual hardware and software items, (2) the ground segment architecture and design will enable the allocated requirements to be met at each level of the specification tree and will support the execution of the user organization's concept of operations, and (3) the ground segment requirements and architecture and design are feasible from a software and hardware perspective. Criteria for evaluation of the products listed in Table 18-1 are found in Appendix D of the *Software Development Standard for Mission Critical Systems* (SDSMCS) [1]. The SDSMCS requires the developer to perform evaluations of the products using the criteria. Some of these criteria are software-specific because they are contained in a standard for software development, but the criteria are easily tailorable to include criteria specific to hardware. In addition, the criteria may be used by the acquisition team as a starting point from which hardware and program-specific software- and hardware-related criteria should be added.

The results of the ground segment requirements development activity are generally reviewed at a system requirements review (SRR), and the results of the ground segment architecture and design are generally reviewed at a system design review (SDR), also known as a system functional review (SFR). In certain circumstances, such as an upgrade to an existing ground segment which does not need to go through the early acquisition phases, these reviews may be combined or held early in the development contract. The list of objectives for

these reviews given in the chapter on readiness reviews may be used to assist in establishing the criteria for reviewing the products of the front-end systems engineering activities. Also of particular usefulness are the standards [6], [7] and [8] because they contain detailed exit criteria for each major technical review.

Table 18-1. Software and Hardware Participation in Front-End Systems Engineering Activities

Activity	Description of the Activity	Products Produced by the Activity
Ground segment requirements development	Participate in defining the ground segment, subsystem, and component functional and non-functional requirements to ensure that software and hardware are properly included in these requirements and to ensure the software- and hardware-related contractual requirements in the Technical Requirements Document (TRD) are properly incorporated into these requirements. [Nonfunctional requirements include dependability (including reliability, maintainability, availability, and safety); human systems integration; supportability (including testability and integrated diagnostics); security (including cybersecurity); performance requirements; and computer resource requirements (e.g., margins). All of these nonfunctional requirements have both software and hardware components.] Also participate in defining the system operations concept to ensure it is feasible with respect to software and hardware. Note that this is an iterative activity that defines the segment, subsystem, and component requirements, as applicable to the ground segment under development.	• Ground segment specification • Ground subsystem and component specification(s), as applicable • Interface specification(s) and interface control document(s), as applicable • Specification tree • Operational concept document (OCD) • Bidirectional traceability among the requirements in the contract TRD and the requirements in the segment, subsystem, and component specifications • Requirements trade studies and engineering analyses for allocation of functional and non-functional requirements to next lower level specifications • Allocation of requirements to software items and hardware items

Activity	Description of the Activity	Products Produced by the Activity
Ground segment architecture and design	Participate in defining the ground segment, subsystem and component architecture and design to ensure that software and hardware are properly included in the architecture and design; that the architecture and design will meet the non-functional requirements as well as the functional requirements; and that, when implemented, the architecture and design will enable the system operations concept to be executed. This includes participating in defining the software and hardware items, their functionality and interfaces, and the allocation of software items to the computer hardware on which they will execute. Note that this activity is generally performed iteratively and concurrently with the requirements development activity described above.	• Segment/subsystem/ component architecture description including use cases and DOD Architecture Framework (DODAF) views • System/segment design document (SSDD) • Segment, subsystem, component, hardware item, and software item architecture trade studies, architecture/design engineering analyses, and make/buy/reuse analyses • Bidirectional traceability among segment, subsystem, and component requirements and their major architectural entities

18.3.1.2 Software Development Activities

Software engineering consists of a set of well-defined activities performed across the lifecycle that, if applied rigorously, provide the "systematic, disciplined, and quantifiable approach" to software development that the definition of software engineering requires. Robust requirements for the software product-oriented activities shown in Figure 18-3 are specified in the SDSMCS [1].

Table 18-2 enumerates the software product-oriented activities, briefly describes each activity, and lists the usual products produced by each activity. The products lead directly to the operational software baseline. It is very important for the acquisition team to thoroughly review these products to ensure the evolving final operational software will meet its allocated requirements and satisfy the user's needs. Criteria for evaluations of the products listed in Table 18-2 are found in Appendix D of the SDSMCS [1]. The SDSMCS requires the developer to perform evaluations of the products using these criteria. These

evaluation criteria, however, may be used by the acquisition team as a starting point from which program-specific criteria should be added.

The results of the software requirements, software architecture, and software qualification test planning activities are generally reviewed at the software architecture and requirements review (SAR) and the preliminary design review (PDR). The results of the software detailed design activity, software integration test planning, and software qualification test case development are generally reviewed at the critical design review (CDR). Readiness for software qualification testing is usually reviewed at a test readiness review (TRR). Also of particular usefulness are the standards from the IEEE and Peresztegy et al., [6–8] because they contain detailed exit criteria for each major technical review.

Software development includes new development, modification, reuse, reengineering, maintenance, and any other activities that result in software products. Most software development is performed in an iterative fashion through a series of "builds," and therefore the software needs to be reviewed on a build-by-build basis. The SDSMCS [1] defines a series of build-level reviews: software build planning review (SBPR), software build requirements and architecture review (SBRAR), software build design review (SBDR), software build test readiness review (SBTRR), and software build exit review (SBER). The criteria for these reviews are found in the chapter on readiness reviews and in Appendix E, of the SDSMCS [1]. These criteria may also be used to assist in establishing the criteria for reviewing the products of software development activities.

Table 18-2. Software Product-oriented Activities

Activity	Description of the Activity	Products Produced by the Activity
Software requirements development	Define the software functional and nonfunctional requirements for each software item based on the ground segment requirements allocated to software, the ground segment architecture and design, and the ground segment operational concept, using the software requirements specification methods and standards documented in the software development plan (SDP).	• Software requirements specifications • Interface requirements specifications • Bidirectional traceability between software requirements, including software interface requirements, and their parent requirements • Software requirements analyses, models, and trade studies

Activity	Description of the Activity	Products Produced by the Activity
Software architecture	Define the software architecture based on the software requirements, ground segment architecture and design, and ground segment operational concept, and using the software architecture methods and standards documented in the SDP. Note that this activity is generally performed iteratively and concurrently with the software requirements development activity described above. Perform initial identification and evaluation of reusable software products (including commercial off-the-shelf, government off-the-shelf (GOTS), open source (OS), and reuse software) for meeting the requirements allocated to the architectural entities, per the SDP.	• Software architecture description (SAD), including architecture diagrams (e.g., unified modeling language [UML®]) and use cases • Bidirectional traceability between the software requirements, including software interface requirements, and the software architecture entities • Software architecture trade studies, engineering analyses, simulations, prototypes, models, and software technology readiness evaluation • Initial COTS, GOTS, OS, and reuse software analyses and decisions
Software detailed design	Define the software detailed design based on the software requirements, ground segment architecture and design, software architecture, and ground segment operational concept, and using the software design methods and standards documented in the SDP. Perform final identification and evaluation of reusable software products (including COTS, GOTS, OS, and reuse software) for incorporation into the operational software, per the SDP.	• Software detailed design, including UML, use cases, class diagrams, data models, user interface design, database design, software interface design, algorithm design, etc. • Software design document (SDD), interface design document (IDD), and database design document (DBDD) • Bidirectional traceability between the software requirements, including software interface requirements, and the software design entities • Software design engineering analyses, simulations, models, prototypes, trade studies, make/buy/reuse analyses, and

Activity	Description of the Activity	Products Produced by the Activity
		software technology readiness evaluation • COTS, GOTS, OS, and reuse software analyses and decisions
Software implementation and software unit testing	Implement the detailed design of each software unit, including coding computer instructions, coding data definitions, building databases, populating databases with data values, creating procedures, populating other data files with values, configuring COTS and any other reusable software, modifying existing code and data, employing automated code generators, and many other tasks needed to implement the design, using the software implementation methods and standards documented in the SDP. Perform unit testing of each implemented software unit, including testing of all algorithms, internal and external interfaces, statements and branches, error and exception handling, nominal and off-nominal conditions, allocated software and interface requirements, fault detection, isolation and recovery, among other unit tests, using the software unit testing methods and standards documented in the SDP.	• Peer reviewed and tested software units, under software configuration management control • Software unit test products and results (plans, cases, procedures, drivers, test data, etc.)
Software unit integration and testing	Integrate the software corresponding to two or more units and perform integration testing of the resulting software, continuing this process until all units are integrated and tested, and	• Integrated and tested software units • Software unit integration and test products and results (plans, cases, procedures, drivers, test data, test databases, build scripts, etc.)

Activity	Description of the Activity	Products Produced by the Activity
	using the unit integration and testing methods and standards documented in the SDP. Integration testing includes testing of all algorithms, end-to-end functional capabilities, internal and external interfaces, concurrent data access, nominal and off-nominal conditions, software and interface requirements allocated to the integrated units including functional and nonfunctional requirements, integrated error and exception handling, and fault detection, isolation and recovery, among other integration tests. Document all discrepancies encountered in this integration testing, whether or not they are immediately resolved.	• Discrepancy reports
Software qualification testing	Perform software qualification testing for each software item to verify that the software item requirements have been met, including all functional and nonfunctional requirements in the software item's software requirements specification and interface requirements specifications, and using the software qualification testing methods and standards documented in the SDP.	• Qualified software items (i.e., integrated software items whose requirements have been verified) • Software test plan, software test description, and software test report for each software item
	Software qualification testing also includes the following types of testing of the entire software item: end-to-end functional capabilities, normal and heavy operational workloads, concurrent data access by multiple users or functions, integrated error and exception handling, worst-case scenarios, endurance	• Other software qualification test products (test data, test databases, build scripts, etc.) • Discrepancy reports

Activity	Description of the Activity	Products Produced by the Activity
	testing, computer resource utilization, and fault detection, isolation and recovery, among other qualification tests. Software qualification testing involves "like you fly testing" of the software in an environment that replicates the operational target computer system, the conditions that the software will encounter in the operational environment (e.g., operational data constants, input and output data rates, and operational scenarios), and the actual interfaces (or high fidelity simulators of the actual interfaces). Software qualification testing includes verification of the software and software interface requirements using all test methods (Inspection, Analysis, Demonstration, and Test) as specified in the Software Requirements Specifications and Interface Requirements Specifications. Software qualification testing includes the test planning, test execution, and analysis of the test results, including documentation of test plans, procedures, results, and all discrepancies encountered.	

18.3.1.3 Hardware Development Core Activities

Table 18-3 enumerates the hardware product-oriented activities, briefly describes each activity, and lists the usual products produced by each activity. The products listed in Table 18-3 lead directly to the operational hardware baseline. It is very important for the acquisition team to thoroughly review these products to ensure the evolving final operational hardware will meet its allocated requirements and satisfy user needs. Additional information on ground segment hardware is found in the chapter on mission management.

The results of the hardware requirements, preliminary design, and hardware verification test planning activities are generally reviewed at the PDR. The results of the hardware detailed design, hardware integration test planning, and hardware verification procedure development activities are generally reviewed at the CDR. Readiness for hardware verification testing is usually reviewed at a TRR. The chapter on readiness reviews has a list of objectives for major technical reviews that may be used to assist in establishing the criteria for reviewing the products of hardware development activities. Also of particular usefulness are the standards from the IEEE and Peresztegy et al. [6–8] because they contain detailed exit criteria for each major technical review.

A standard on ground segment testing, *Test Requirements for Ground Systems* [9] is recommended for use on ground segment development contracts. This standard addresses hardware level testing as well as higher-level testing of ground systems.

Table 18-3. Hardware Product-oriented Activities

Activity	Description of the Activity	Products Produced by the Activity
Hardware requirements development	Define the hardware functional and nonfunctional requirements for each hardware item based on the ground segment requirements allocated to hardware, the ground segment architecture and design, and the ground segment operational concept.	• Hardware requirements specifications, including hardware interface requirements • Bidirectional traceability between hardware requirements, including hardware interface requirements, and their parent requirements • Hardware engineering analyses, simulations, models, prototypes, and trade studies
Hardware preliminary design	Define the hardware preliminary design to the level of hardware units and their interfaces, based on the hardware requirements, ground segment architecture and design, and ground segment operational concept. Perform initial identification and evaluation of COTS and other reuse hardware products for incorporation into the hardware items and units.	• Top-level hardware item drawings to the level of hardware units and their interfaces • Bidirectional traceability between the hardware requirements, including hardware interface requirements, and the hardware units • Engineering analyses, simulations, models,

Activity	Description of the Activity	Products Produced by the Activity
		prototypes, make-buy-reuse analyses, and hardware technology readiness evaluation • Initial layout drawings for equipment in operational facilities, and initial analyses of ability to meet operational site constraints (e.g., footprint; heating, ventilation, and air conditioning (HVAC); power, and weight) • Initial COTS and reuse hardware analyses and decisions
Hardware detailed design	Define the hardware detailed design to the level of hardware units, assemblies, subassemblies, and parts as appropriate, based on the hardware requirements, hardware preliminary design, ground segment architecture and design, and ground segment operational concept. Perform final identification and evaluation of COTS and other reuse hardware products for incorporation into the hardware items and units. Prepare procurement specifications for procuring COTS hardware products. For hardware items or units incorporating COTS hardware or COTS integrated hardware and software products, the hardware detailed design is generally not elaborated below the COTS entities and their interfaces.	• Detailed hardware item drawings of hardware units, assemblies, subassemblies, parts, and their interfaces, as appropriate • Bidirectional traceability between the hardware requirements, including hardware interface requirements, and the hardware units, assemblies, and subassemblies • Engineering analyses, simulations, models, prototypes, make-buy-use analyses, and hardware technology readiness evaluation • Layout drawings for equipment in operational facilities, and analyses of ability to meet operational site constraints (e.g., footprint, HVAC, power, and weight) • COTS and reuse hardware analyses and decisions • Procurement specifications for COTS hardware products and COTS integrated hardware and software products

Activity	Description of the Activity	Products Produced by the Activity
Hardware Fabrication/ Procurement & Hardware Unit Testing	Fabricate hardware being developed for this ground system, building up from parts to subassemblies to assemblies, as designed. Modify reuse hardware as designed for incorporation into the hardware units. Procure COTS hardware and perform acceptance testing upon delivery to ensure it functions as specified. Integrate COTS, reuse, modified reuse, and newly developed hardware into subassemblies, assemblies, and units per the design. Test all subassemblies, assemblies, and units to ensure they function as designed.	• Hardware units, tested and under configuration management control • Hardware subassembly, assembly and unit test products and results (e.g., plans, procedures, test environments, test data, results)
Hardware Item Integration and Testing	Integrate two or more hardware units and test the integrated units until the entire hardware item is integrated and tested. Each unit may contain or consists of COTS hardware product(s), reuse hardware, modified reuse hardware, and newly developed hardware. Perform integration testing to ensure the integrated units function as designed. Document all discrepancies encountered in this integration testing, whether or not they are immediately resolved.	• Integrated and tested hardware items under configuration management control • Hardware item integration and test products and results (e.g., plans, procedures, test data, test environments, results) • Discrepancy reports

Activity	Description of the Activity	Products Produced by the Activity
Hardware Requirements Verification	Perform hardware requirements verification testing for each hardware item to ensure that the hardware item requirements have been met, including all functional and nonfunctional requirements in the hardware item's hardware requirements specification and including all hardware interface requirements. Hardware verification testing includes the following types of testing of the entire hardware item: end-to-end testing, testing under normal and heavy operational workloads, concurrent access by multiple users or functions (as applicable), worst-case scenarios, endurance testing, and fault management (fault detection, isolation and recovery), among other hardware verification tests. Hardware verification testing involves "like you fly testing" of the hardware in an environment that replicates the conditions under which the hardware will operate, using the actual hardware interfaces (or high fidelity simulators of the actual interfaces) and operational data rates. Hardware verification testing includes verification of the hardware and hardware interface requirements using all test methods (Inspection, Analysis, Demonstration, and Test) as specified in the hardware requirements specifications and interface requirements specifications or interface control documents. Hardware verification testing includes the test planning, test execution, and analysis of the	• Hardware items whose requirements have been verified, under configuration management control • Hardware item verification testing products and results (e.g., plans, procedures, test data, test environments, results) • Discrepancy reports

Activity	Description of the Activity	Products Produced by the Activity
	documentation of test plans, procedures, results, and all discrepancies encountered. Document all discrepancies encountered in this integration testing, whether or not they are immediately resolved.	

18.3.1.4 Component and Subsystem Integration and Verification Core Activities

In addition to integration and test personnel, development team and acquisition team personnel who are experts in the software and hardware disciplines involved in the ground segment development must participate in the higher-level integration and verification activities. These higher-level integration and verification activities consist of ground segment component and subsystem integration and verification activities, as applicable to the specific ground segment under development. Detailed requirements on all levels of ground segment testing are addressed by Lutton [9]. This standard is recommended for use on all ground segment development contracts.

For each of these component and subsystem integration and verification activities, Table 18-4 briefly describes the activity and lists the usual products produced by the activity. The phrase "participate in" is used in the description of the activity since the activity is commonly the responsibility of integration and test engineering personnel, and software and hardware development personnel take part in that activity.

The products listed in Table 18-4 lead directly to the operational component and subsystem baselines. Thus, it is very important for the acquisition team to thoroughly review these products to ensure the final operational components and subsystems will meet their allocated requirements and satisfy the user's needs. Criteria for evaluations of the products listed in Table 18-4 are found in Appendix D of the SDSMCS [1]. The SDSMCS requires the developer to perform evaluations of the products using these criteria. These evaluation criteria may be used by the acquisition team as a starting point from which program-specific criteria should be added.

Readiness for component and subsystem verification testing is usually reviewed at a TRR. The list of objectives for this review may be used to assist in establishing the criteria for reviewing the products of the component and subsystem verification activities. Also of particular usefulness are the standards

from the IEEE and Peresztegy et al. [6–8] since they contain detailed exit criteria for this major technical review.

Table 18-4. Component and Subsystem Integration and Verification Core Activities

Activity	Description of the Activity	Products Produced by the Activity
Component and subsystem integration and testing	Participate in integration of one or more software and hardware items in each ground segment component, and participate in performance of integration testing of the integrated software and hardware items, continuing this process until all software and hardware items in each component are integrated and tested. Participate in integration of one or more components, hardware items, and software items in each ground segment subsystem, as applicable, and participate in performance of integration testing of the integrated components, hardware items, and software items until all components, hardware items, and software items in each subsystem are integrated and tested. This integration testing includes testing of all end-to-end functional capabilities, internal and external interfaces, concurrent access, nominal and off-nominal conditions, fault management, and requirements allocated to the components and subsystems including functional and nonfunctional requirements, among other integration tests. Participate in documenting all discrepancies encountered in this integration testing, whether or not they are immediately resolved.	• Integrated and tested ground segment components and/or subsystems, under configuration management control • Integration test products and results (e.g., plans, procedures, test data, test environments, results) • Discrepancy reports • Qualified/verified software and hardware items, updated to repair discrepancies found by component and subsystem integration and testing

Activity	Description of the Activity	Products Produced by the Activity
Component and subsystem requirements verification testing	Participate in performing component and subsystem requirements verification testing for each ground segment component and subsystem to ensure that its requirements have been met, including all functional and nonfunctional requirements in the component's or subsystem's requirements specification and including all interface requirements. This verification testing includes the following types of testing of the entire component or subsystem: end-to-end testing, testing under normal and heavy operational workloads, concurrent access by multiple users or functions (as applicable), worst-case scenarios, endurance testing, and fault management (fault detection, isolation and recovery), among other tests. Component and subsystem verification testing involves "like you fly testing" of the component or subsystem in an environment that replicates the conditions under which it will operate, using the actual interfaces (or high fidelity simulators of the actual interfaces) and operational data rates. The verification of the component or subsystem requirements includes using all test methods (Inspection, Analysis, Demonstration, and Test) as specified in the component and subsystem requirements specifications and their associated interface requirements specifications or interface control documents. This verification testing includes the test planning, test execution, and analysis of the test results, including documentation of test plans, procedures, results, and all discrepancies encountered.	• Ground segment components and/or subsystems, whose requirements have been verified, under configuration management control • Component and subsystem verification testing products and results (e.g., plans, procedures, test data, test environments, results) • Discrepancy reports • Qualified/verified software and hardware items, updated to repair discrepancies found by component and subsystem requirements verification testing

18.3.1.5 Software and Hardware Participation in Ground Segment Test Planning and Execution

In addition to ground segment test personnel, development team and acquisition team personnel who are experts in the software and hardware disciplines involved in the ground segment development must participate in the ground segment test planning and execution activities to ensure that (1) the ground segment test planning and test procedures correctly reflect the functioning of the hardware and software items in the ground segment, (2) the hardware and software items operate correctly in the fully integrated environment, and (3) all discrepancies encountered in the ground segment testing are properly documented with enough information that the discrepancy can be replicated and repaired.

The activities of ground segment test planning and execution include integration of the ground segment subsystems and verification of the ground segment requirements, including end-to-end testing of the entire ground segment. There is also a standard on ground segment testing, [9], which contains detailed requirements on ground segment testing. This standard is recommended for use on all ground segment development contracts.

The operational versions of the hardware and software items result from updating the hardware and software items to repair discrepancies found by the ground segment level testing. The operational versions of the components and subsystems are created from the operational versions of the hardware and software items. These are the versions of the hardware and software items, components, and subsystems that transition into operations and maintenance.

18.3.1.6 Hardware and Software Preparation for Transition to Operations and Maintenance

Hardware and software preparation for transition to operations and maintenance must occur throughout the development lifecycle in order for the ground segment to be ready for these transitions when they must occur. It is important for acquisition team hardware and software personnel to monitor the contractor's transition preparation activities and participate in reviewing the operations and maintenance products for correctness and completeness so that accurate products are produced for the eventual operators, end users, and maintainers of the ground segment and so that the ground segment is fully ready for the start of operations and maintenance.

18.3.1.6.1 Preparation for Transition to Operations

Software preparation for transition to operations includes:

- Preparation of the plan for software transition to operations. This may be included in a ground segment plan or in a software-specific plan for transition to operations.
- Preparation of executable software for each operational site, prepared from configuration managed source code
- Procurement of sufficient COTS software licenses for each operational site
- Preparation of a software version description document for each operational site that contains the description of the software being delivered, installation instructions, and description of any known problems in the software with applicable workarounds
- Installation and checkout of executable software and COTS software at each operational site. Checkout may include executing part or all of the software, component, or subsystem verification tests.
- Preparation of software user manual(s) providing the operators for all operator positions and the end users with instructions on operating the software. Currently, these manual(s) are provided in an on-line format available to the operators and end users.
- Procurement of COTS software manuals and training as needed for operators and end users
- Participation in preparation of software training materials for end users and operators for all operator positions. These training materials may be provided on-line or through a training system.
- Participation in preparation and checkout of operations procedures (work instructions) for all operator positions
- Participation in population of operational databases and other data files

Adams et al. [1] address software preparation for transition to operations to include criteria for reviewing some of the software operations products. These criteria can be used as a starting place for developing criteria for reviewing the software operations products.

Hardware preparation for transition to operations includes:

- Preparation of the plan for hardware transition to operations. This may be included in a ground segment plan or in a hardware-specific plan for transition to operations.
- Procurement of sufficient COTS hardware for all operational sites
- Procurement of COTS hardware manuals and training as needed for operators and end users
- Installation and checkout of developed and COTS hardware at each operational site. Checkout may include executing part or all of the hardware, component, or subsystem verification tests.

- Preparation of instructions for operating the hardware for each operator position. Currently, these instructions are provided in an on-line format available to the operators.
- Preparation of instructions for end users to operate the hardware, if end users are to operate any of the ground segment hardware. These instructions are generally in an on-line format available to the end users.
- Participation in preparation of hardware training materials for operators for all operator positions and end users (if applicable). These training materials may be provided on-line or through a training system.
- Participation in preparation of hardware operations procedures (work instructions) for all operator positions

18.3.1.6.2 Preparation for Transition to Maintenance

Software preparation for transition to maintenance includes:

- Preparation of the plan for software transition to maintenance. This may be included in a ground segment plan or in a software-specific plan for transition to maintenance.
- Procurement of sufficient COTS software licenses for the maintenance site, both for the operational software and for the software maintenance and test environments at the maintenance site. Note that this includes the COTS software for the firmware maintenance environment, if applicable.
- Procurement of hardware for the software maintenance and test environments for the maintenance site. Note that this includes the hardware for the firmware maintenance environment, if applicable.
- Installation and checkout of the hardware and software in the maintenance and test environments, ensuring that the hardware and software in these environments perform their intended functions. This includes the hardware and software in the firmware maintenance environment, if applicable.
- Procurement of COTS software manuals for the maintenance environment, both for operational COTS software and COTS software in the maintenance and test environments, including the firmware maintenance environment.
- Preparation of the source code baseline for delivery to the maintenance site
- Preparation of the executable software from the source code baseline for delivery to the maintenance site
- Preparation of software version description for the maintenance site that contains the description of the source code and executable software being delivered, installation instructions, and description of any known problems in the software with applicable workarounds

- Installation and checkout of source code, executable software, and operational COTS software in the maintenance environment. Checkout may include execution of part or all of the software, component, or subsystem verification procedures.
- Update of the software requirements, architecture, and detailed design to ensure they reflect the as-built software. Delivery of these materials and installation into the maintenance site.
- Preparation and delivery of the software product specification. This is the usual mechanism for the acquirer to receive the source code and executable code baselines, together with the as-built software requirements, architecture, and detailed design and the software maintenance instructions.
- Delivery of the software version descriptions for all operational sites to the maintenance environment to ensure the maintainers know the exact version of software installed in the operational sites and the procedures used to install that software
- Delivery of unit test, software integration test, and software qualification test products, including regression testing products, to the software maintenance environment so that the maintainers do not have to develop new testing products from scratch.
- Delivery of other information and products from the development environment to the maintenance environment that will be useful to the maintainers (e.g., software development files).
- Preparation of software maintenance manuals for information needed by the maintainers that is not contained in COTS software manuals. These manuals are generally provided in an on-line format available to the software maintainers in the software maintenance environment.
- Preparation of detailed software maintenance procedures for maintaining the source code, databases, and data files and creating new baselines of executable software for the operational sites. These procedures are generally provided in an on-line format available to the software maintainers in the software maintenance environment.
- Demonstration that the software source code can be maintained by the maintainers in the maintenance environment. This includes creation of the executable software from the source code baseline delivered to the maintenance site and comparison of the created executable software with the executable software delivered to the maintenance site to ensure that exactly the same executable software was created in the maintenance environment as was delivered to the maintenance environment from the development environment. It also includes demonstrating the correct performance of the software maintenance processes for diagnosing software problems (e.g., from discrepancy reports), changing the software source code to fix the problems, recreating a new set of executable software, installing that software in

the target hardware in the test environment, and testing that software to ensure that the problems were correctly fixed.

- Preparation of detailed firmware maintenance procedures for updating the software portion of the firmware, downloading that software in the firmware device, and testing the new version of the firmware to ensure it functions as required, if the ground segment includes developed firmware.
- Demonstration that the software portion of firmware can be maintained by the maintainers and that a new version of the firmware can be created in the firmware maintenance environment and tested to ensure the new firmware executes as required, if the ground segment includes developed firmware.
- Participation in preparation and checkout of software maintenance work instructions used by the maintainers
- Participation in preparation of software maintenance training materials for the software maintainers. This includes firmware maintenance training materials, if applicable to the ground segment. These training materials may be provided online or through a training system.

Adams et al. [1] address software preparation for transition to maintenance and criteria for reviewing some of the software maintenance products. These criteria can be used as a starting place for developing criteria for reviewing the software maintenance products.

Hardware preparation for transition to maintenance includes:

- Preparation of the plan for hardware transition to maintenance. This may be included in a ground segment plan or in a hardware-specific plan for transition to maintenance.
- Preparation of logistics analyses for hardware support at the operational sites and at the off-site hardware maintenance depots.
- Procurement of the needed number of spare line replaceable units (LRUs) for the operational sites and spare parts, subassemblies, assemblies, and LRUs for the hardware maintenance depots.
- Procurement of COTS hardware maintenance manuals and training for on-site hardware maintainers and for maintainers at the hardware maintenance depots.
- Preparation of hardware maintenance procedures for the operational sites and at the hardware maintenance depots. These procedures are generally in an on-line format available to the hardware maintainers on-site and at the hardware maintenance depots.
- Participation in the preparation of hardware training materials for on-site hardware maintainers and hardware maintainer in the depots. These training materials may be provided online or through a training system.

- Participate in ensuring the hardware on-site and depot maintenance are functioning properly to ensure the operational hardware remains in an operational state.

18.3.2 Integral Core Activities

As shown in Figure 18-4, there are four categories of integral activities: project management, process management, quality enhancement, and specialty engineering. While all of these activities are important for a successful ground segment development project, this handbook addresses in detail only those integral activities related to ground segment systems engineering. However, it is important for the acquisition team to be aware of all these activities, to ensure that they are being performed correctly, and to review the products of these activities when appropriate.

18.3.2.1 Project Management Activities

The project management activities that apply to ground segment product development include the following:

- Planning: This activity involves developing the plans for the ground segment development, including quantitative planning (e.g., cost, schedule, and staffing); qualitative planning (e.g., processes, methods, techniques, procedures, standards, and tools); and planning for necessary facilities containing robust development and test environments, with sufficient resources to support the project schedules. The results of the planning are documented in the integrated management plan, systems engineering management plan, software development plan, hardware development plans, COTS hardware and software procurement plans, and other plans as needed.
- Monitoring and controlling: This activity includes monitoring the ground segment development against the plans and performing corrective action as required.
- Management reviews: This activity includes holding joint management reviews between the contractor team and the acquisition team, ensuring that the management review objectives are met. The objectives of management reviews generally include reviewing status of the ground segment development against the plans, resolving issues, and agreeing upon risk mitigation strategies. There are many types of joint-management reviews including program management reviews, integrated product team meetings, monthly and weekly hardware and software reviews with the contractor and acquisition team managers and technical experts in the specific subject areas.
- Measurement and analysis (metrics): This activity involves planning, collecting, analyzing, and reporting metrics data for the ground

segment development effort and integrating the metrics data and analysis into the monitoring and controlling activity. More information can be found in the chapter on metrics.

- Risk management: This activity involves identifying, analyzing, handling, reporting, and tracking to closure ground segment development risks, including ground segment systems engineering, hardware development, software development, and segment integration and verification risks. Risk handling includes developing strategies for managing or mitigating the risks and implementing those strategies. The risk management activity also includes elevating lower-level risks (e.g., hardware and software risks) to the program level risk management process when the risks cannot be handled strictly within the lower level organization. More information can be found in the chapter on risk assessment and management.
- Corrective action: This activity involves implementing a closed-loop corrective action system that ensures each detected discrepancy is documented, reported, evaluated, dispositioned, fixed as applicable, and tracked to closure. This activity includes all discrepancies, not just those found in testing; it includes discrepancies found in products and processes as well.
- Configuration and data management: The configuration management activity involves performing configuration identification, configuration control, configuration status accounting, and configuration audits for all entities placed under configuration control, including products and elements of the ground segment development and test environments. Configuration management also includes baseline management. The activity of data management involves establishing and maintaining a repository of all technical and management data produced by the ground segment program throughout the lifecycle. Data management includes providing the capability for easy access of all data in the repository by both the development team and acquisition team. More information is found in the chapter on configuration and data management.
- Subcontractor management: This activity involves ensuring all ground segment contractual requirements are properly flowed down to all subcontracts or work authorizations for all contractor team members. It also includes monitoring subcontractor performance and performing corrective action when that performance deviates from contractual or work authorization requirements.

18.3.2.2 Process Management Activities

The process management activities that apply to ground segment product development include the following:

- Process definition: This activity involves defining the processes to be used on the ground segment development effort for the technical and management activities throughout the lifecycle. This activity also includes defining the detailed work instructions, procedures, and standards to be used by the ground segment development personnel and making the processes, work instructions, procedures, and standards available to the appropriate technical and management personnel on the project.
- Process improvement: This activity involves assessing the processes in use on the program for suitability, efficiency, and effectiveness, including the work instructions, procedures, and standards. It also includes planning and executing process improvement to make beneficial improvements to the processes and correct any deficiencies found.
- Training: This activity involves provide training to all ground segment development technical and management personnel on the processes, work instructions, procedures, and standards to be used on the program.

Process management began with the software discipline, but now addresses all systems engineering, hardware development, software development, and integration and verification activities. See CMMI® [3] for more details.

18.3.2.3 Quality Enhancement Activities

The quality enhancement activities that apply to ground segment product development include the following:

- Technical reviews: This activity involves preparing for and executing all types of technical readiness reviews and ensuring that predefined entry and exit criteria are met.
- Product evaluations: This activity involves performing in-process and final evaluations of products produced during the development lifecycle (as listed in Tables 18-1 through 18-4), using a pre-established set of evaluation criteria.
- Peer reviews: This activity involves performing peer reviews on the work products produced throughout the lifecycle, using predefined checklists for each type of work product. Performing peer reviews includes planning the review, preparing for the review, conducting the review (includes identifying and documenting the discrepancies found), and analyzing and reporting the data. Peer reviews have been proven to be extremely effective in reducing defects in the evolving products throughout the development lifecycle.
- Quality assurance: This activity involves conducting ongoing evaluations of (1) adherence of the product-oriented and integral activities to the documented processes and to the contract and (2)

adherence of the products being produced to the documented standards and the contract. The quality assurance chapter has more detail on this subject.

18.3.2.4 Specialty Engineering Activities

Specialty engineering activities are performed concurrently with the product-oriented activities and influence the products produced by those activities. It is essential that specialty engineering requirements be implemented into the ground segment as development proceeds, through all levels of the ground segment: subsystems, components, and software and hardware items.

This handbook addresses those specialty engineering areas most important to ground segment systems engineering in the chapters on the following topics.

- Reliability, maintainability, availability (RMA) and fault management
- Fault management
- Cybersecurity
- Human systems integration
- System safety

18.3.3 Key Lessons Learned for Ground Segment Product Development

The following product development lessons learned apply to ground segment acquisitions:

- A robust software development standard should be used for all software in the ground segment development. The SDSMCS [1] is a robust software standard that addresses all product-oriented and integral software development activities and is recommended for use as a compliance document on all ground segment development contracts.
- A robust ground segment test standard, such as [9], should be used for all ground segment development acquisitions. This standard is recommended for use as a compliance document in all ground segment contracts.
- The evolving ground segment hardware and software development is reflected in the products listed in Tables 18-1 through 18-4. It is extremely important that these products be available for acquisition team review in a timely manner. It is recommended that an electronic access clause be included in all ground segment development contracts that requires the contractor (1) to electronically store all technical and management data in a form easily accessible by both the contractor team and acquisition team, and (2) to store the data in a timely manner on the electronic access system.

- Some products should be required as deliverable to the government (CDRL) because of their importance to a successful ground segment acquisition. Software products that are recommended for delivery are described in Owens and Tagami [10]. Updated instructions for the contents for some of these products are found in Appendix H of the SDSMCS [1]. Systems engineering products important to successful software development are described by Owens and Tagami [11]. It is especially important to ensure that the CDRL items include the source code itself, since otherwise the contractor is not obligated to deliver the source code to the acquirer. The usual mechanism is to have the software product specification as a CDRL item since this document contains the source code baseline, executable code baseline, as-built software requirements, architecture, and detailed design, and software maintenance instructions.
- Data rights for the software are an especially important issue for ground systems, which may be in sustainment for decades. The government must have the proper rights to the software so that the acquirer may compete future software maintenance to reduce costs. The development contract must specify the required data rights for all categories of software, including newly developed, reuse, modified reuse, COTS, and open source software. This includes requirements on the types of licensing and other restrictions of the vendors for COTS and other types of software (e.g., no embedded expiration dates where the COTS product will stop working if a license expires without renewal). Additional information on data rights is found in Abelson et al. [12]. Additional information on COTS software is found in Adams and Eslinger [13].
- While the data accession list (DAL) is an essential deliverable for contract management, it is not an effective mechanism for obtaining development technical and management products in a timely manner. Generally it takes too long to obtain products from the DAL for acquisition team review comments to be effective in improving the quality of the products. The products need to be deliverable or available in a timely manner on an electronic access system for effective review.
- Additional information on incorporating software into the request for proposal (RFP) is found in Abelson et al. [12]. Information on incurporating software assurance into the RFP is found in Eslinger et al. [4].

The following product development lessons learned apply to contract monitoring of the evolving ground segment product development throughout the lifecycle:

- Technical review teams should be established for reviewing the ground segment development products. These teams must have the necessary breadth and depth of technical knowledge and experience to be able to review the products effectively and efficiently. Product reviews should

always be based upon a set of criteria established by the review team before the review starts. Product review comments must be delivered to the contractor on schedule in order for the contractor to be able to improve the products based on the review team's work.

- Product review is necessary, but not sufficient, for ensuring that the evolving ground segment will meet the contractual requirements and the user's needs. Simulations, models, and prototypes are needed to demonstrate that the ground segment, software, and hardware architectures and designs will meet their requirements. As an example, ground segment performance based on the architecture should be modeled using a dynamic simulation technique to demonstrate that the architecture will satisfy the performance requirements under the operational workload. The simulation should be refined as the hardware and software architecture and design are developed in more detail throughout the life cycle. As another example, ground segment reliability modeling, including modeling of the software reliability, should be performed to show that the ground segment, hardware, and software architecture and design will meet their RMA requirements. If this type of simulation, modeling, or prototyping is not being done by the contractor, the acquisition team should arrange for it to be done by The Aerospace Corporation or other government-support contractor.
- Participation in technical meetings and reviews should involve technical personnel with expertise in the subject matter, not just management or program office personnel who do not have the necessary expertise to understand whether or not the material presented by the contractor is acceptable from a technical perspective. This includes all types of readiness reviews discussed in the chapter on the topic.
- The acquisition team must ensure that testing is monitored by technical personnel with the necessary breadth and depth of knowledge and experience to be able to understand the execution of the evolving system and the problems encountered. For software, monitoring of testing should begin no later than build integration and test and must include software item qualification testing. Similarly, for hardware, monitoring of testing should begin during hardware item integration and testing and must include hardware verification testing. Component and subsystem integration testing and verification testing should always be monitored.
- Maintainability must be built into the hardware and software as the development work proceeds. Building software maintainability into the software through the requirements, architecture, detailed design, code, integration and test, and qualification testing activities is discussed in Adams et al. [14]. Maintainability criteria should be included when reviewing the evolving ground segment products throughout the lifecycle.

- Security must also be built into the hardware and software as the development work proceeds, to ensure the risk of exploitable vulnerabilities is sufficiently small. Building security into the software through the development lifecycle is discussed in Eslinger et al. [4]. Security criteria should always be included when reviewing the evolving ground segment products throughout the lifecycle.

18.4 Software and Hardware Development Lifecycle Models

A lifecycle model is a project management framework providing a sequencing strategy and a disciplined approach to the structure and order of activities. This section describes software development and hardware development lifecycle models.

18.4.1 Waterfall Lifecycle Model

The foundational lifecycle model for both hardware and software is the waterfall or "once through" lifecycle model, depicted in Figure 18-5. In the waterfall model, the basic product-oriented activities for hardware or software development occur only once, in a sequential order. The names of the activities in Figure 18-5 are the names of the software product development activities shown in Figure 18-3 and described in Table 18-2. For hardware, replace the names of the activities in Figure 18-5 with the names of the hardware product development activities shown in Figure 18-3 and described in Table 18-3.

Figure 18-5 depicts the activities as overlapping and with some iteration between the successive activities. In the early uses of this lifecycle model, the phases did not overlap and there was no feedback between successive activities. Over the years, however, it became understood that overlap and feedback were necessary for successful development using this lifecycle model.

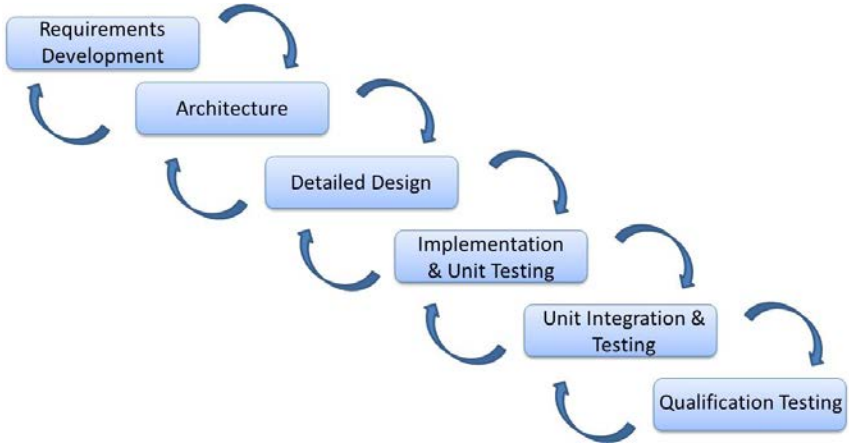


Figure 18-5. Example waterfall lifecycle model.

Figure 18-6 depicts the placement of a software waterfall lifecycle model into the ground segment life cycle. The ground segment life cycle shown at the top of Figure 18-6 may be interpreted as the entire ground segment development as in a waterfall ground segment lifecycle, or it may be interpreted as a single ground segment increment as in an iterative ground segment lifecycle.

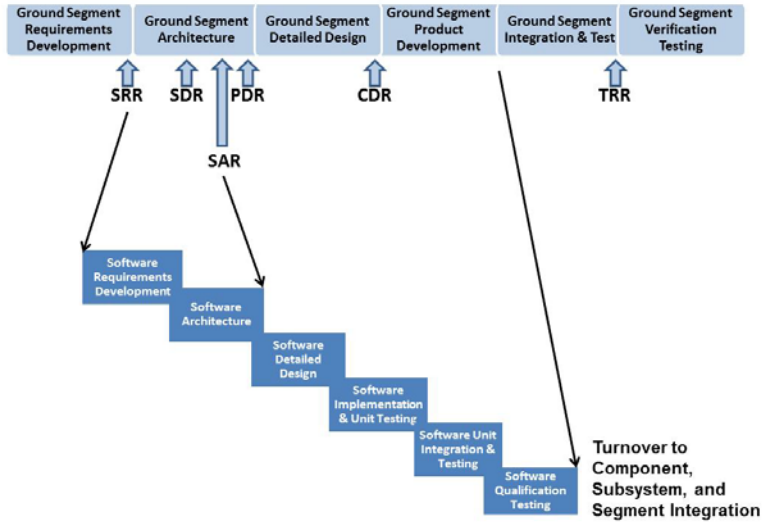


Figure 18-6. Software waterfall lifecycle to ground segment lifecycle mapping.

Hardware development generally follows the waterfall lifecycle model. To depict the placement of a hardware development waterfall lifecycle model into the ground segment lifecycle, replace the software product development activities in Figure 18-6 with the hardware product development activities. The same alignment with the ground segment lifecycle applies to both hardware and software.

Figure 18-7 depicts a mapping between the software product development activities and the principal software products produced by those activities for the waterfall lifecycle model. Also shown on Figure 18-7 is a mapping between the software build reviews and the software product development activities. Note that a waterfall lifecycle model can be thought of as having one build, which is the entire software. (Builds are discussed in more detail in the next section, and software build reviews are described in more detail in the chapter on readiness reviews.) A mapping similar to Figure 18-7 can be produced for the principal hardware products for a hardware waterfall lifecycle model.

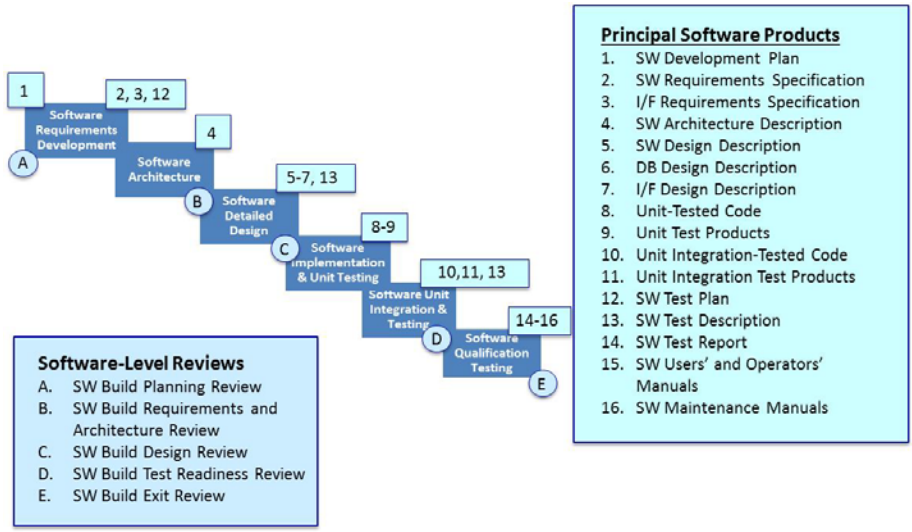


Figure 18-7. Mapping of software products and reviews to the software waterfall lifecycle model.

In ground segment development there are usually multiple hardware and software items to be developed, as shown in the ground segment decomposition example in Figure 18-2. Figure 18-8 shows ground segment development with three software items and three hardware items where all of the hardware and software items are developed using the waterfall lifecycle model. In reality, ground segment development is much more complex than shown in this figure because it usually involves many more hardware and software items. The

difficulty of using the waterfall lifecycle model for both hardware and software comes because each hardware and software item is developed independently after the ground segment architecture and design activity, following the allocation of requirements to the individual hardware and software items. Integration of the individual hardware and software items does not occur until after the hardware and software item development is complete, and thus integration problems are not found until very late in the ground segment development when such problems are difficult, expensive, and time consuming to correct. This type of problem led to the development of iterative software development lifecycle models, where early integration of the software with the hardware could occur.

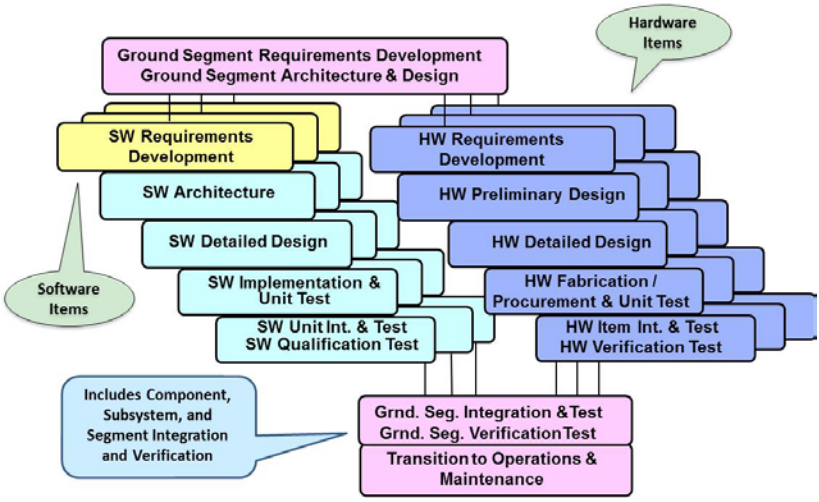


Figure 18-8. Waterfall lifecycle model with multiple software and hardware items.

18.4.2 Iterative Software Development Lifecycle Models

Software development is now understood to be best accomplished using an iterative lifecycle model unless the amount of software to be developed is quite small. Most ground segment software development, therefore, follows an iterative lifecycle model because the amount of software in a ground segment is usually quite large, running into millions of source lines of code (SLOC).

In iterative software development lifecycle models, the software is developed in a series of iterations, usually called builds.

There are many types of iterative software development lifecycle models. This section will describe two of the models frequently used in ground segment

development are the incremental and agile lifecycle models. More information about software development lifecycle models is found in Holloway et al. [15].

18.4.2.1 Incremental Software Development Lifecycle Model

Figure 18-9 depicts an incremental software development lifecycle model. In the incremental software development lifecycle model, the software requirements for all software items are developed up front, and then are allocated to builds. Instead of the software items being developed independently, builds contain parts of one or more software items that integrate together. Each build integrates with the preceding build and adds capability until all of the software items are completely developed and tested.

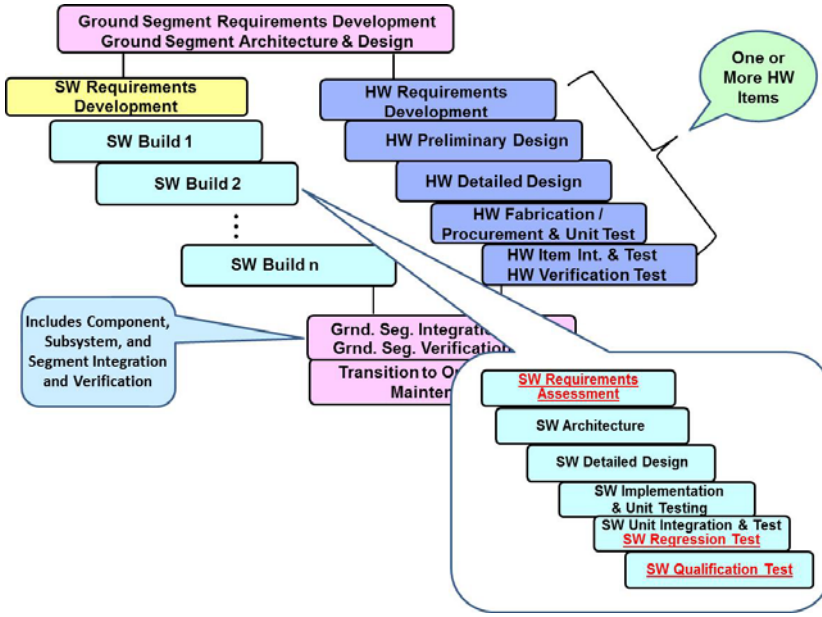


Figure 18-9. Example incremental software development lifecycle model.

The development of each build is a mini-waterfall, as shown in Figure 18-9. Since the software requirements have already been developed, the build starts with a software requirements assessment activity to assess and update the software requirements allocated to the build for any changes that have occurred since the up-front software requirements development activity. The build then goes through the usual sequence of software architecture, detailed design, implementation and unit testing, and unit integration and testing. The unit integration and testing activity integrates the software developed for this build

with all of the previous builds. After the first build, regression testing must be performed following the build's integration and testing activity to ensure that the build has not affected any of the capabilities developed and tested in previous builds.

If the build is to be delivered outside of the software organizations (e.g., to the ground segment integration and test organization), then software qualification testing is usually performed for that build. However, if the build is strictly internal to the software organization, software qualification testing may be omitted from the build and performed during a later build that is transitioning outside the software organization. Some versions of the incremental software development lifecycle model perform software qualification testing only at the last build. At that time, the software qualification testing verifies all of the software requirements, not just those allocated to a single build.

The principal advantage of the incremental software development lifecycle model is that it enables early integration across software items and early integration with the computer hardware on which the builds will operate. As the builds proceed, the software test environment must become more and more equivalent to the eventual operational environment. Thus, the early builds may undergo integration testing and regression testing in a test environment that contains the target hardware, but the test environment may be limited in the quantity of the equipment available and the ability to use real interfaces with other hardware and software, including external interfaces. However, by the later builds, the software should undergo integration and testing, regression testing, and qualification testing in the ground segment test facility, which should contain a representative replica of the operational environment.

18.4.2.2 Agile Software Development Lifecycle Model

There are many different agile development methods that have been defined during recent decades. Most promote development, teamwork, collaboration, and process adaptability throughout the life cycle of the program.

Agile methods break tasks into small increments with minimal planning. Each increment may contain multiple iterations. Iterations are short time frame developments (timeboxes) that typically last from one to four weeks, usually called sprints. Each sprint employs a cross functional team performing software planning, development, and test activities, and each sprint delivers a working product (either internally or externally). Also, each agile team contains a customer representative whose role is to be available for developers to answer questions between deliveries and to make decisions. A common characteristic of agile development is daily team status meetings where team members report to each other what they did the previous day, what they learned, what they intend to do today, and what their roadblocks are.

Agile software development involves activities, similar to waterfall and other iterative software development lifecycle models, but the activities are organized around sprints. Agile development begins with a planning sprint where the software requirements are translated into features and the features are converted to "user stories" with associated acceptance criteria. This planning session lays out the software architecture, determines the number of sprints, and allocates the user stories to the sprints. At the beginning of each development sprint is another planning session where the user stories assigned to the sprint are decomposed into tasks and the tasks are assigned to team members. During the development sprints the detailed design, implementation and unit test, and integration and test activities occur. These activities can occur on any day and different parts of the software will be in a different activity at different times. At the end of each day, the software that has completed integration and test is deployed internally to the rest of the team so that each developer is building and testing to the latest version of the software. Software qualification testing; component, subsystem and ground segment integration and testing; and deployment follow the end of the development sprints.

Figure 18-10 shows an example of an agile software development lifecycle mapped to the ground segment development lifecycle. The production sprints shown in the figure are for maintenance of the release that has been turned over to an organization external to the software development for the various levels of integration and test. These sprints would occur on an as-needed basis.

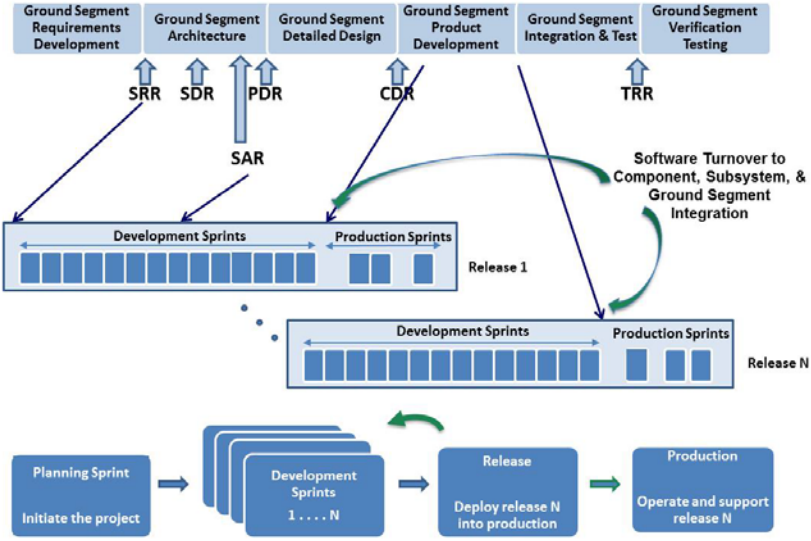


Figure 18-10. Example mapping of agile development lifecycle to ground segment lifecycle.

The principal advantage of agile development methods is that they easily accommodate the changing needs of the users and changes to the higher-level requirements allocated to software. They also allow the software architecture and design to evolve as changes occur. Like the incremental software development lifecycle model described above, agile methods also allow for early integration of the software with the target hardware, and as the sprints proceed, the software test environment must become more and more equivalent to the eventual operational environment.

18.4.3 Key Lessons Learned for Lifecycle Models

The following lessons learned for lifecycle models apply to ground segment acquisitions:

- The government should not specify the development lifecycle model(s) to be used for the hardware or software product development. The contractor should be allowed to choose the lifecycle model(s) based on their corporate processes and experience. The government should review and approve the contractor's choice of lifecycle model(s) to ensure their choice is appropriate for the ground segment hardware and software products being developed.
- For software, the contract should require compliance with a robust software development standard that accommodates iteration, various types of software development lifecycle models, and software build-level reviews, such as by Adams et al. [1].
- When traditional iterative lifecycle models (e.g., incremental and evolutionary) are used, creation of the principal software products occurs build by build, and the products are not generally completed until the last build. Delivery of software CDRL items (e.g., software requirements specifications, software architecture description, software test plans) should occur on a build-by-build basis consistent with the contractor's software development lifecycle model. Owens and Tagami [10] provide sample language for multiple deliveries of software CDRL items for traditional iterative software development lifecycle models.
- Agile software development is very different from traditional software development. If an offeror bids using agile development on a ground segment development contract, the software products and reviews must be tailored for content and schedule consistent with the offeror's agile development lifecycle, if that offeror wins the award. Similarly, the software build reviews will need to be tailored for content and schedule for an agile software development contract. Writing the RFP properly to allow both traditional and agile lifecycle models to be bid by different offerors is difficult and will require participation of experts in software acquisition.

- More information on preparing an RFP for a ground segment where the software is expected to be developed in iterations is found in Abelso et al. and Holloway et al. [12] and [15].

The following lessons learned for lifecycle models apply to contract monitoring of the evolving ground segment product development throughout the lifecycle:

- Criteria for reviewing the development products must be established on an iteration-by-iteration basis. The software development products, especially, will evolve as the software development lifecycle proceeds. For each iteration the acquisition team must have software technical experts available to review the software products and participate in the software build reviews.
- Effective monitoring of agile development involves embedding software personnel with sufficient software expertise to review the evolving software product and provide guidance concerning the user's viewpoint to the developers into the contractor's agile teams. Software acquisition team personnel with expertise in the particular type of agile development in use are also needed.
- Review of the technical products by acquisition team personnel will be more effective if the acquisition team has the same tools available that the contractor is using. This enables the acquisition team engineering personnel to perform analysis using the capabilities built into the tools.

18.5 References

1. Adams, R. J., et al., *Software Development Standard for Mission Critical Systems*, TR-RS-2015-00012, The Aerospace Corporation, El Segundo, CA. March 17, 2014.

2. https://en.wikipedia.org/wiki/Agile_software_development 2015

3. CMMI Product Team, *CMMI® for Development, Version 1.3 (CMMI-DEV, V1.3)*, Carnegie Mellon University (CMU)/Software Engineering Institute (SEI), CMU/SEI-2010-TR-033, November 2010.

4. Eslinger, Suellen, et al., *Integrating Software Assurance into the Request for Proposal*, TOR-2013-00694, The Aerospace Corporation, El Segundo, CA. January 31, 2014.

5. International Organization for Standardization/International Electrotechnical Commission /Institute for Electrical and Electronic Engineers (ISO/IEC/IEEE), *Systems and Software Engineering—Vocabulary*, ISO/IEC/IEEE Standard 24765:2010, 15 December 2010.

6. IEEE, *Draft Standard for Technical Reviews and Audits on Defense Programs*, IEEE P15288.2/D5.2, September 2014.

7. Peresztegy, L. B. and C. E. O'Connor, *Technical Reviews and Audits of Systems, Equipment, and Computer Software*, TOR-2007(8583)-6414, Rev. 1, Vol. 1, The Aerospace Corporation, El Segundo, CA. January 30, 2009.

8. Peresztegy, L. B. and C. E. O'Connor, *Technical Reviews and Audits of Systems, Equipment, and Computer Software*, TOR-2007(8583)-6414, Rev. 1, Vol. 2, The Aerospace Corporation, El Segundo, CA. January 30, 2009.

9. Lutton, David, et al. *Test Requirements for Ground Systems*, TR-2013-00215, The Aerospace Corporation, El Segundo, CA. June 4, 2013.

10. Owens, Karen L., and J. M. Tagami, *Recommended Software-Related Contract Deliverables for National Security Space System Programs*, TOR-2006(8506)-5738, The Aerospace Corporation, El Segundo, CA. February 14, 2008.

11. Owens, Karen L., and J. M. Tagami, *Recommended Software-Related Systems Engineering Contract Deliverables for National Security Space System Programs*, TOR-2005(8506)-8101, The Aerospace Corporation, El Segundo, CA. June 27, 2008.

12. Abelson, Linda, A., et al., *Integrating Software Topics into the Request for Proposal,* TOR-2011(8506)-117, The Aerospace Corporation, El Segundo, CA. July 19, 2012.

13. Adams, R. J. and S. Eslinger, *COTS-Based Systems: Lessons Learned from Experiences with COTS Software Use on Space Systems*, TR-2001(8550)-1, The Aerospace Corporation, El Segundo, CA. September 2001.

14. Adams, Richard J. and S. Eslinger, *Software Sustainment Guidance*, TOR-2013-00693, The Aerospace Corporation, El Segundo, CA. December 31, 2013.

15. Holloway, Leslie J., et al., *Iterative Software Development in Space Systems Acquisition*, TOR-2013-00692, The Aerospace Corporation, El Segundo, CA. June 1, 2014.

18.6 Bibliography

Institute for Electrical and Electronic Engineers (IEEE), *Draft Standard for Application of Systems Engineering on Defense Programs*, IEEE P15288.1/D4.1, September 2014.

International Standards Organization (ISO) / International Electrotechnical Commission (IEC) / IEEE, *Systems and software engineering – System life cycle processes*, ISO/IEC/IEEE 15288-2008, 21 March 2008.

Eslinger, Suellen, L. J. Holloway, and R. M. Wilkes, *Space Segment Software Readiness Assessment,* TOR-2011(8591)-20, The Aerospace Corporation, El Segundo, CA. June 3, 2011.

Guarro, S. et al., *Mission Assurance Guide (MAG)*, TOR-2007(8546)-6018 Rev. B, The Aerospace Corporation, El Segundo, CA. June 1, 2012.

Unell, Alan, et al., *Evaluating Software Architectures in Space and Ground Systems*, ATR-2012(9010)-12, The Aerospace Corporation, El Segundo, CA. 2012.

Department of Defense (DOD), *Operation of the Defense Acquisition System*, Department of Defense Instruction DODI 5000.02, 7 January 2015.

18.7 Acronyms

CDR	critical design review
CDRL	contract data requirements list
CMMI®	Capability Maturity Model® Integration℠
COTS	commercial off-the-shelf
CSCI	computer software configuration item
DAL	data accession list
DB	database
DBDD	database design document
DOD	Department of Defense
DODAF	DOD architecture framework
FFRDC	Federally Funded Research and Development Center
GOTS	government off-the-shelf
HVAC	heating, ventilation, and air conditioning
HW	hardware
I/F	interface
IDD	interface design document
IEC	International Electrotechnical Commission
IEEE	Institute for Electrical and Electronic Engineers
IRS	interface requirements specification

ISO	International Organization for Standardization
LRU	line replaceable unit
OCD	operational concepts document
OS	open source
PDR	preliminary design review
RFP	request for proposal
RMA	reliability, maintainability, and availability
SAD	software architecture description
SAR	software requirements and architecture review
SBDR	software build design review
SBER	software build exit review
SBPR	software build planning review
SBRAR	software build requirements and architecture review
SBTRR	software build test readiness review
SDD	software design document
SDP	software development plan
SDR	system design review
SDSMCS	software development standard for mission critical systems
SE&I	systems engineering and integration
SETA	systems engineering and technical assistance
SFR	system functional review
SLOC	source lines of code
SM	service mark
SRR	system requirements review
SRS	software requirements specification
SSDD	system/subsystem design document
STD	software test description
STP	software test plan
STR	software test report
SW	software
TRD	technical requirements document
TRR	test readiness review
UML®	unified modeling language

Chapter 19
Ground Segment Test Planning and Execution

Gregory Lockwood
Software Systems Assessment
Software Systems Assurance Department

19.1 Introduction

There are steps required to plan and execute an end-to-end test of the entire ground segment. These steps include the planning for robustness and regression tests, in addition to planning for development of operational test scenarios. Descriptions are provided of the tasks needed to integrate the various hardware and software subsystems into a complete ground segment to include verification of requirements at the system level by both test and analysis. In addition, the tasks are described associated with sell off of requirements to the customer, as well as the verification that all requirements have been allocated down to the appropriate level, including validation activities.

The verification process begins early, and continuing throughout the lifecycle, and includes planning, requirement definition, and compliance. The highest level of verification activities occur prior to launch through to the operation phases. Post-launch verification activities may occur due to software updates and other changes to the ground segment.

A rigorous verification process is essential to ensuring the successful acquisition of reliable and cost-effective ground systems. General principles provide a roadmap and methodology for developing and executing a verification plan. This guidance is primarily intended for use by the government and contractors who are directly involved with the acquisition of the ground segment. The guidelines can be tailored to fit specific mission needs and ground configurations. The verification approach descriptions provide details on the requirement verification analyses and test.

Ground segment verification should address both design verification and system requirement verification. Design verification ensures a given design meets the system requirements. Emphasis is placed on mission assurance, such as performance and reliability with a risk reduction focus, to ensure the hardware and software built meet requirements. Engineering models and simulators are key tools for design verification that are focused on critical requirements and parameters to meet customer needs. System requirement verification considers: acceptance testing (screening built items for material and workmanship defects); software item qualification testing for software maturity; and verification and validation of the simulators and database.

Hardware or software failures during the ground integration and test phase at the factory or at the ground site usually result in significant cost and schedule impacts. Failures on-orbit can result in the partial or full loss of the entire mission, or possibly endanger the health and safety of the spacecraft. The goal of the verification process is to verify requirements at the lowest possible level of integration. This verification process will prevent a significant number of the failures in the higher levels of integration and test (I&T) by detecting problems early enough to prevent significant program impacts further downstream.

The ground test requirements are applicable to the design, development, and integration of new, reused, or modified software and hardware in ground systems that provide command and control, mission management, mission data processing, and common services (e.g., data storage, networks, and other shared functions) for space vehicles. Requirements also apply to training systems and simulators used for training operators and maintainers of ground systems.

19.2 Definitions (1)

Acceptance The action taken by the acquirer's designated representative by which the acquirer assumes ownership of products that signify partial or complete performance of a contract.

Acceptance test The required formal tests conducted to demonstrate acceptability of an item for delivery. The tests are designed to demonstrate performance to specified requirements and to act as quality control screens to detect deficiencies of workmanship, material, and quality.

Build A version of software that contains a configuration controlled set of capabilities, software changes, or fixes.

Commercial off-the-shelf (COTS) Commercial items that require no unique government modifications or maintenance over the lifecycle of the product to meet the needs of the procuring agency. A COTS item (hardware, software, or both) is produced and made commercially available or placed in stock by a vendor prior to the vendor receiving orders or contracts for sale of the item.

Component A generic designation for a part of the system (not used to describe a level of software hierarchy, since that use is obsolete, nor as a synonym for a hardware part).

Defect A flaw in a system, hardware, or software item. The cause of a failure. That which is corrected to correct a failure. Also called fault.

Developer The groups or organizations performing the development of the ground system, including prime contractor, subcontractors, team members, government development staff, and/or associated agencies' development staff.

Discrepancy A functional or structural anomaly or failure which indicates a possible deviation from specification requirements for the test item. A test discrepancy may be a momentary, nonrepeatable, or permanent failure to respond in the predicted manner to a specified combination of test environment and functional test stimuli. Test discrepancies may be due to a failure of the test item or to some other cause, such as the test setup, test instrumentation, supplied power, or test procedures.

Discrepancy Report (DR) Description of the discrepant behavior and effect on the system component being tested. A report may also include information for managing and analyzing defects and testing, such as proposed correction, due date, assignee, completion date, source (requirements, design, implementation, etc.) or cause of the discrepancy. Also called problem report, deficiency report, and other similar terms.

Failure The inability of a system or component to perform its functions as specified or within specified performance parameters. Behavior of a system or system component that deviates from specified requirements. A test step not passed.

Firmware Firmware is a combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device. The software cannot be readily modified under program control.

Functional Configuration Audit (FCA) Verifies that all item or subsystem requirements established in the functional and allocated baselines, specifications, and test plans have been verified successfully, and corrective action has been initiated, as necessary.

Integration Connection and/or interoperation of newly developed or modified software or hardware test items into configurations that allow test of functions and operations of the combined items. Also connection and/or interoperation of the ground system with other items, existing or in development, e.g., spacecraft or other ground systems.

Physical Configuration Audit (PCA) Physical examination of the actual configuration of the item produced. It verifies that the design and product documentation specified in the contract matches the as-built item and that all documentation/deliverables to the operational site are present, complete, and accurate.

Procuring Activity The procuring activity is the government office or agency with primary responsibility for developing and acquiring the system, subsystem, equipment, computer software, or engineering services addressed in this document.

Regression Test A test performed after system changes have been made to verify that the changes did not inadvertently introduce failures to system functions that were working properly prior to the changes.

Requirements Verification Traceability Matrix (RVTM) See Verification Cross-reference Matrix.

Software Development Environment The computers and software development tools used to conduct coding and unit testing of a software item. The computers and support software may not be identical to the target hardware and support software.

Software Development File (SDF) A repository for material pertinent to the development of a particular body of software. Contents typically include (either directly or by reference) considerations, rationale, and constraints related to requirements analysis, design, and implementation; developer internal test information; and schedules and status information.

Software Integration and Test Environment The developer-controlled and developer-maintained configuration of computer and communications hardware, test and measurement software, and equipment used to perform integration and qualification testing of software.

Specification A document used in development and procurement that describes the technical requirements for items, materials, and services. Specifications may be unique to a specific program (program peculiar) or they may be common to several applications (general in nature).

Subassembly The term subassembly denotes two or more parts joined together to form a stockable unit which is capable of disassembly or part replacement. Examples are a printed circuit board with parts mounted, or a gear train.

Subsystem, Hardware A subsystem is an assembly of two or more components, including the supporting structure to which they are mounted, and any interconnecting cables or tubing. A subsystem is composed of functionally related components that perform one or more prescribed functions.

Subsystem, Software Synonymous with, or instead of, software item in some programs. The specification level consisting of software requirements.

System of Systems A set or arrangement of interdependent systems that are related or connected to provide a given capability. The loss of any part of the system will significantly degrade the performance or capabilities of the whole.

System Verification Review (SVR) A multidisciplinary technical review to ensure the system is ready to proceed into low-rate initial production and full-rate production within cost (program budget), schedule (program schedule), risk, and other system constraints. Generally this review provides an audit trail from the critical design review.

Test Activities conducted to obtain data to verify that an implementation is as-designed, that specification requirements are satisfied, and to identify defects and deficiencies for corrective action. Tests that verify these requirements can use verification methods described in the definitions for: verification by analysis; analysis by demonstration; analysis by inspection; and verification by test. The narrower use of the term 'test' is one method to verify a requirement, as described in the definition for test, verification by.

Test Documentation File A repository for material pertinent to the test of a unit, subsystem, or other component of a ground system. Contents typically include (either directly or by reference) test plans, procedures, data, results, reports, and schedules and status information. Would also include analysis of anomalies, and root cause of anomalies, rationale for retest, and regression test.

Test Environment Test environments for hardware consist of the facility housing the equipment under test, necessary fixtures or special handling equipment, plus the required test instrumentation. Test environments for software consist of the computing hardware running the software, test drivers, databases and simulators, data collection and analysis software, test execution scripts, or automated test software.

Test Item Levels The levels used in this document, from the simplest to the most complex, are:

Hardware	Software
Part	Unit
Subassembly	Software Item
Unit	Subsystem (alternate for software item or not used in some systems)
Subsystem	System
System	

Additional levels such as “segment” and “element” are used in large ground “system-of-systems configurations.” These levels will be defined by the procuring activity.

Test Method Standard A standard that specifies procedures or criteria for measuring, identifying, or evaluating qualities, characteristics, performance, and properties of a product or process.

Test Readiness Review (TRR) A multidisciplinary technical review and process assessment to ensure that a subsystem or system is ready to proceed into formal test. The TRR assesses test objectives, test methods and procedures, scope of tests, and safety, and confirms that required test resources have been properly identified and coordinated to support planned tests.

Unit, Hardware A separately testable component of a hardware design that is part of a subsystem.

Unit, Software An element in the design of a software item; for example, a major subdivision of a software item, a component of that subdivision, a class, object, module, function, routine or database. Software units in the design may or may not have a one-to-one relationship with the code and data entities (routines, procedures, databases, data files, etc.) that implement them or with the computer files containing those entities.

Validation Confirmation that the product or service, as provided (or as will be provided), will fulfill its intended use. In other words, validation ensures that “you built the right thing.”

Verification Confirmation that work products properly reflect the requirements specified for them. In other words, verification ensures that “you built it right.”

Verification Cross-reference Matrix (VCRM) Section of each specification or a separate document that lists, for each requirement, the requirement identifier and the verification method for the requirement—inspection, analysis, demonstration, or test; referred to as RVTM on some programs.

19.3 Verification Planning and Execution Considerations

A well-run program begins the verification planning process early in the program requirements definition and risk reduction phase rather than waiting until after the preliminary design review.

19.3.1 Verification Planning

The planning process is a critical and iterative process that should be closely coupled with the overall design strategy and schedule planning for a program. The verification strategy should diligently assess and plan for mitigation of the risks identified as part of the design approach. For example, planning should acknowledge the extended development schedule, budget, and testing usually needed for new technology applications. The method, level (e.g., segment or system), approach, and success criteria for assessing design and as-built requirement compliance should be defined as part of the verification planning. Multiple verification objectives include first level inspection to include units, subsystems (e.g., thermal vacuum environmental testing); second level verification at system level; and third level verification of on-orbit test. Requirements are mapped to the effectiveness based on: capabilities being delivered; maturity of the ground segment software; survivability and endurability of the mobile ground elements; maturity of interfaces; maturity of factory support systems; and the space segment maturity. Design and requirement verification reports leverage the contractor processes and documentation to include test plans, procedures, and results.

The process for final buyoff of the system requirements should use the lower-level verification artifacts from the unit and subsystem specifications to support satisfaction of the system-level requirements. This documentation approach produces a traceable verification process. The process of using lower-level requirements to partially or fully verify higher-level requirements is sometimes known as “rollup”. A draft verification plan should be produced at system design review (SDR), with an update at preliminary design review (PDR) and the final plan completed at the critical design review (CDR).

19.3.2 Organizational Accountability

The program system engineering customer and contractor organization have the responsibility of requirement verification buyoff. Those organizations, in turn, look to the ground segment functional design areas to demonstrate that the allocated requirements are satisfied by analyses and test. The verification plan is documented at lower levels, and summarized in a specification cross-reference paragraph of each specification. This methodology encourages verification documentation at the lowest levels and, if done correctly, minimizes rework costs to the program while providing a reliable verification approach.

A verification integrated product team (IPT) with customer and contractor members should be initiated in the early phases of the program to ensure appropriate and applicable development and execution of verification plans and processes. The customer and the contractor must continuously participate in the IPT in order to fully engage in the multitude of verification-related activities,

and to quickly resolve both programmatic and technical risks when identified. An IPT will normally address the higher level activities such as interaction with the space segment or the simulated spacecraft, whereas smaller working groups (WG) may be formed to perform activities for lower systems such as subsystems and units. An IPT/WG will:

- Participate in the development of upper-level specification verifications, such as those relating to the system, ground, and the space-to-ground interface. Participate in the mission-requirement flow-down process that will be coordinated by the top-level system engineering IPT.
- Participate in the development of the internal interfaces (I/Fs), and lower-level specifications while ensuring the proper requirement flowdown from these specifications. Also, the team will ensure that the required "derived/housekeeping" requirements are properly included in each of the related specifications. Examples of derived requirements may be those levied to the ground hardware control software to support an antenna's pointing accuracy, whereas housekeeping requirements may be those associated with the storage of archived data.
- Develop an overall ground segment verification plan, and I&T plans that are consistent with the upper-level system and interface verification plans.
- Facilitate the proper assignment of verification methods to each requirement of the end-to-end specifications so that the individuals and/or combinations of lower-level verifications will sufficiently verify the higher-level requirements.
- Ensure development and execution of the verification is initiated in the early phases of the program. This process should be refined and continued throughout the life of the program from SDR, PDR, CDR, and ground I&T.

During the verification planning phase, experienced representation from the hardware and software testing organizations will be intimately involved with the verification. Experts in system analysis responsible for mission performance requirements throughout the lifecycle are also contributors directing system test running scenarios that include tactical, strategic, and stressing day-in-the-life scenarios. Contributions from the working group focused on critical requirements such as radiometric calibration and command plan generation ensure adequate testing to include review of the fidelity of the simulators and use of validated data and test environments. Assuring the design is right mandates that the verification of the design, especially on highly complex systems, addresses all anticipated worst-case scenarios. The goal is to prove that

the system can withstand the stressing conditions that it may encounter during operations.

19.3.3 Verification of Requirements

System performance, environmental, functional, design, production, operability, and interface requirements are captured in the system technical requirements baseline. One of the most critical aspects of developing requirements is to assure every requirement is unambiguous and verifiable. Common occurrences exist throughout the industry where failure to do this has caused significant schedule delays and trouble actually verifying the requirement. One of the most extreme cases of flawed verification planning is in a situation where a customer is unclear or rushed into agreed design based on capability rather than a set of specified requirements.

Figure 19-1 illustrates that verification should occur at each level. Higher-level requirement specifications flow down to lower-level specifications, and verification is developed and performed at the lowest level first, and then flowed up to the higher levels.

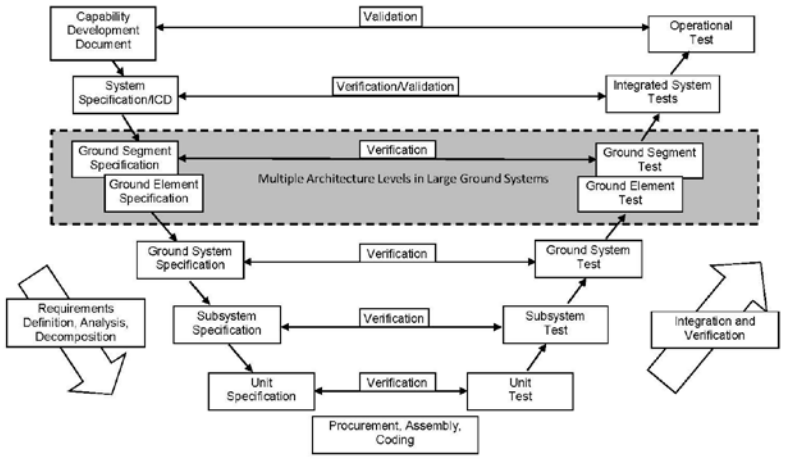


Figure 19-1. Requirements test diagram [1].

Successful ground systems must have a rigorous requirements flow-down process to the unit level. A comprehensive verification process ensures each requirement is properly verified at the lowest possible level of integration. By using this verification philosophy, program cost overruns and schedule slips are minimized because problems are caught as early as possible in the development phase of the program. Test failures at the unit level are much faster and cheaper to fix than test failures at the subsystem or system level. Verification IPT/WGs

must participate in the requirement flow-down and the specification development processes in conjunction with the development of each of the system or segment specifications and associated verification plans.

Each requirement of a specification must be written such that it is objectively verifiable by any one or combination of the verification methods explained here. Requirements must be achievable and testable; Words such as “any”, “all”, “always”, and “never” should be used with extreme caution in requirement text, as the use typically leads to requirements which are difficult or impossible to verify. The system and lower-level specifications contain not only mission requirements, flowed-down from higher-level specifications, but also “derived” or “housekeeping” requirements that are unique to each segment, element, subsystem, unit, or associated interface. Specifications must avoid subjective statements. For example, qualitative statements such as “single-point failures must be avoided as much as possible” are subjective and not verifiable. Requirements should always be stated in an objective and verifiable way. The previous example would meet these criteria if it were rewritten to state, “a single-point design shall not be allowed to degrade the mission unless it is identified in a critical failure modes list.”

Verification activities include: (1) establishment of system- and lower-level specifications that are objectively verifiable, (2) development of system end-to-end (system to unit) verification plans, including the assignment of appropriate verification methods to each requirement of the specifications, and (3) execution and documentation of these plans. Integration and test plans are developed as a subsidiary document to the system verification plan.

19.3.3.1 The Verification Plan

The verification plan is a required deliverable document and included on a contract deliverable requirements list (CDRL) along with the I&T plan. Lower-level verification and I&T plans should also be included in the CDRL list, especially if these elements are to be contracted out to subcontractors. This requirement should be included in the statement of work (SOW) as well as the integrated master plan (IMP). The verification program should be flowed down from the prime contractor to the subcontractors and vendors to ensure the verification program plans are consistent with the prime contractor’s program and management processes.

The verification plan describes the contractor system verification activity as well as the contractor’s interaction with the customer system program office. The approach includes a description of the delivery of reports, usually as part of a requirements verification ledger for review by the customer.

19.3.3.2 Setting Up a Tracking System

The documentation required for verification coordinates the many people and activities involved throughout a ground system acquisition. An accounting procedure is needed to assure compliance of verification requirements, which translates into some sort of a traceability matrix, such as a VCRM. For each requirement, the method of verification is maintained, as well as identifying the documentation of the verification products, such as test results and reports that detail the results of any analyses performed. Figure 19-2 is an example of one line from a VCRM at the unit level. The resolution of problems found during the design and verification activities may result in changes in requirements and in design, and the tracking system must be responsive to these changes. Some of these will require a change in the requirement or the verification approach to include re-verification as needed.

Req ID	Req Text	Method *	Level	Verification Description	Verification Products
ADPE123	The Automated Data Processing Equipment (ADPE) servers shall be capable of failing over to a redundant server in less than 15 minutes.	T	Subsystem	While executing the system at full operational load, simulate a failure on an ADPE server. Complete a failover of the server to a backup within 15 minutes.	Testcase Server_Failover, results in Test Report located in directory <xyz>

*D: Demonstration, I: Inspection, A: Analysis, T: Test

Figure 19-2. An example of a unit-level VCRM.

19.3.3.3 Developmental Tests

Developmental tests are planned and performed by the developers to verify correct implementation and compliance with specifications, provide evidence for acceptance of developed systems, and collect data to certify readiness for operational testing.

Unit-level testing provides the initial insight into performance, verifies the design meets requirements (discovery of design errors), and provides an effective screen for workmanship. A unit is subjected to environmental tests to include thermal cycle and vibration. Software is integrated and tested at the unit level as well. Requirements for rigor in testing at the box or unit level is necessary to ensure interfaces to the subsystem and eventually the segment and system are verified incrementally. Subsystem level of integration includes all hardware and software/firmware required to make it work. Redundancy management is a consideration to ensure that reliability, availability, and

maintainability requirements are met. Subsystem testing may be accomplished with similators to ensure the configured subsystem is tested to verify all functions work as specified and desired.

The criticality of the understanding of mission operations is required to define the verification activities required. Developmental tests are conducted to validate new design concepts and technology, reduce risk in the design and build, validate qualification and acceptance test procedures and investigate concerns, problems, or anomalies. Types of developmental tests include:

- **Development tests.** Tests conducted during development of hardware and software items to confirm design adequacy and verify that hardware manufacture and assembly, and software implementation perform as designed.
- **Qualification tests.** Tests performed to demonstrate that hardware, software, or integrated items meet specified requirements.
- **Acceptance tests.** Formal tests required for contractual acceptance of a ground system.
- **Integrated system tests.** Tests performed with the new ground system integrated with other program elements or segments under development, and with existing installed systems, to verify correct interface implementation and correct end-to-end operation of the integrated systems.

19.3.3.4 System Integration Tests

Integrated system tests are performed to exercise, to the maximum extent that is practical and possible, the system in development plus all systems that interface and interoperate with that system. Unless exceptions are approved by the procuring activity, integrated system tests are performed on the integrated hardware and software items installed in an operational system with connection to actual interfaces, and these tests are conducted at the target site with the support of the operational personnel.

The objective of the tests is to ensure that the products, even if from multiple developers, function correctly when integrated, that interfaces are verified, and that all system requirements or specifications are met. Tests are designed to use actual mission operations configurations and conditions to the maximum practical extent. End-to-end tests are performed that exercise the full operational configuration, including space vehicle (possibly simulated), with operational timelines and data loads. External systems that could affect the operation of system(s) in test, e.g., RF emitters, are operated or simulated to replicate conditions expected during operations. Test articles, configurations and

conditions that differ from the operational configuration shall be identified in test planning and risk mitigation.

A development test bed approved by the procuring activity as sufficiently simulating the operational system capability for test purposes may be used for integrated system tests if target sites, operational complexes, or other suitable operational support areas are not available.

The integrated system tests shall incorporate tests of the affected interfaces of the ground equipment and software with other elements of the operational system. The tests shall be structured as appropriate to demonstrate design requirements of the system related to such items as performance, electromagnetic compatibility, reliability, maintainability, system safety (e.g., hazardous noise, radiation hazards, pressure vessels), logistics supportability, operational procedures, and personnel performance.

The tests demonstrate the following, as applicable to the installation, modification, or upgrade:

- Reliable operation is achieved at specified design limits
- System functional and performance requirements are met
- System can recover from hardware or software malfunctions within the required time without excessive loss of data or control
- Performance requirements are met under required logical or physical device assignment combination
- Software and hardware modifications or upgrades have not degraded the capability of the system's baseline or of other operational systems
- Security mechanisms are in place or incorporated to protect resources from unauthorized access or break-in

Tests are to be focused on the external interfaces involved, the use of operational databases and operational scenarios/loads, and the system requirements from a mission operations perspective.

19.3.3.5 Operational Tests

Operational tests are conducted by the procuring activity or a designated test organization, in coordination with the operations organization. Test support is provided by developers as designated in the SOW. These tests are conducted to test and evaluate the capability of the ground system to meet the operational requirements. Operational tests may be combined with developmental tests when objectives can be satisfied concurrently.

19.3.3.6 Retest and Regression Test

Retesting is performed if a test discrepancy or test item failure occurs while performing any of the required testing. Following correction of the problem, a retest of the failed test case and related areas identified by analysis of the implemented fix is performed. When analysis shows previous tests have been invalidated by the failure, those tests are repeated. Regression testing of software is performed after any modification to previously tested software, and after each installation of software in a new environment or at a new site to identify unexpected side effects of software changes.

19.3.3.7 Verification Methods

Verification methods include test, demonstration, analysis, and inspection. This order is generally from most to least rigorous. Verification can be performed by a single method or a combination of these methods. Assignment of these different methods depends on whether or not the system is a new, heritage, modified-heritage system, or heritage but applied to a new environment. The selection and assignment of these different methods must be determined by the appropriate IPT.

Test. Test is the actual operation of equipment or software in an operational environment, where measurements are taken for the verification of requirements. The "verification by test" method can be performed in subsystem tests as well as during the various stages of ground integration and testing. Verification using a method of "Test" typically includes recording and/or analyzing data, for example verification of performance, thresholds, or margins. An example of verification by test would be using system logs to calculate the latency to send a command to the spacecraft, and to verify that it meets the performance requirement.

Demonstration. Demonstration is the use of the system in the manner it was intended in order to prove that it can satisfy the requirement. The difference between this method and test is that demonstration is typically verified by simply observing the appropriate reaction on the running system, without the need for measurement or analysis. An example of this would be executing the ground software in conjunction with the commanding and telemetry (C&T) hardware to prove that the system has a visual indication when the command link is down.

Analysis. The "verification by analysis" method may be accomplished based on any one, or a combination of, mathematical or statistical methods, modeling and simulation, prototypes, or using data from software and hardware engineering documents produced during product development or provided by COTS vendors. In some cases, analysis may need to utilize the results of a separate test.

The selection of specific analytic approaches to verify a specific requirement must be coordinated in the appropriate IPTs. The analysis documentation number must be referenced in the appropriate section of the VCRM. An example would be verifying that the ground system has an availability of 99.5%. It would be impractical to verify this requirement by executing the system. The preferred method would be to analyze the network architecture, along with inputs from the equipment vendors on mean time between failures (MTBF) and mean time to recover (MTTR) to calculate the expected availability using analysis.

Inspection. Inspection is the physical evaluation of equipment and/or documentation to verify design features. The "verification by inspection" is normally applied to checking of such items as the physical characteristics of ground hardware, inspection of vendor provided documentation, or review of design documents.

Other methods. A few texts recognize other methods such as similarity, simulation, and validation of records. These can be considered as subsets of the four methods defined above.

The choice of verification methods is a potential source of risk. Use of inappropriate methods can lead to inaccurate verification. The preferred method is to verify a requirement is met by actual test. Often it is not possible to test a parameter that corresponds to a top-level requirement. In these cases, subordinate parameters can usually be tested and the results combined analytically to verify compliance with the top-level requirement. However, in some cases, it will not be possible to directly measure any parameters, in which case analytic means must be relied on. There may also be practical limitations to the preferred testing method—e.g., verifying that the disk storage can hold 30 days of operational data. This could be instead verified by using other test results for data growth over a certain period of time, and then using analysis to determine how much data would be generated over the required time.

Special consideration should be given to heritage hardware (HW) and software (SW). It is sometimes assumed that no additional verification effort is required for such items. This is rarely the case, as even minor modifications or differences in applications, interfaces, and environments can have large, unforeseen, and potentially disastrous consequences. An example would be modifications that do not change the operational software, but may change database items. It may not be obvious that additional verification is needed, but the database changes could cause a major impact if the software is unable to correctly process the new values. The proposed verification effort for heritage items should be confirmed by critical analysis and review.

19.3.4 Expected Products per Development Phase

Concept Development

- Note any aspects of the concept that relate to performance verification and assess the risk to the program.

System Requirements Review

- Requirements should be verifiable.
- Description provided for the overall structure and objectives of the test and evaluation program, including use of modeling and simulations, schedule, and required resources.
- Necessary developmental test and evaluation activities identified.
- Test objectives related to critical operational issues, critical technical parameters, minimum acceptable performance requirements, evaluation criteria, and exit criteria established.
- Initial test and evaluation planning conducted.
- Preliminary requirements compliance matrix (requirement vs. capability matrix) generated.

System Design Review

- Preliminary I&T plan and analysis plan, from unit to system level, of acceptable risk and credible schedule with reasonable margins created.
- Preliminary segment verification plan under configuration management (CM).

Preliminary Design Review

- Final system verification and I&T plans released and under CM.
- Final segment verification plan under CM.
- Preliminary segment and element I&T plans under configuration control.

- Key verification analyses are completed and show that the design meets the requirements with adequate margin.

- Preliminary plans are established to support I&T (down to assembly level), verification (down to subsystem level and unit as appropriate), and early on-orbit and continued operations.

- Preliminary I&T plan should be completed. The plan delineates what analyses and tests will be performed over the life of the program to ensure a quality product that meets requirements. PDR requirements will be satisfied when the document is completed down to the subsystem level.

- For each subsystem the required analyses and tests to verify the design should be established. The required analyses and test for design verification must be presented in preliminary verification plans and matrices or ledgers. The verification plan should be defined down to the subsystem and unit level. The end-to-end plan for verification and validation of the design should be defined. The applicability of heritage item qualification and qualification by similarity should be defended, and supporting data provided. Analyses and test plans should be provided. Plans should provide for unit test implementation to ensure minimum returns at the subsystem and system level. Test plans must address unit, subsystem, and system-level testing.

- SW development processes should be described. SW qualification approaches should be defined. Schedules for design, build, verification, and test of SW through final build and integrated HW/SW simulation testing/verification should be delineated.

Critical Design Review

- Verification plans should be updated and complete, including system verification details. I&T plans to build, assemble, integrate, test, and validate system complete from unit to system level test and launch site testing.

- Analyses are complete and show that the design meets the requirements with adequate contingency and margin, e.g., network bandwidth central processing unit (CPU) processing, and disk storage space.

- Detailed system I&T plans, procedures, and schedule with critical path analysis complete.

- Design performance across internal system interfaces using mission operations concept threads demonstrated.

- For each subsystem analyses and test plans should be final. Verification matrices/ledgers should be complete except for planned events. The verification plans should be final. The end-to-end plan for verification and validation of the design should be complete. The applicability of heritage item qualification and qualification by similarity should be defended with supporting data provided.

- SW development plans, status of requirement flow-down to computer SW modules, and detail plans and schedules for SW build and test completed. Rationale/justification provided to show SW test methodology using HW and final simulation verification in exercising the HW and SW in a valid HW-in-the loop and simulation environment, to provide confidence that the computer and SW are tested for functional/operational requirement performance over the life of the vehicle. Validation of all SW builds defined. SW and HW qualification approach validated.

System Validation and Acceptance

- System completely verified.

- System end-to-end performance demonstrated. This demonstration should include the complete set of systems that the system will interface with.

19.4 Key Lessons Learned

The lessons learned, and re-learned, over the course of numerous programs have given rise to good organizational rules and practices related to qualification and acceptance test strategy. These practices should be included in the verification planning and procedures. The practices below are a good starting point, but are not a substitute for qualified experienced verification and test personnel.

- Start the verification program at the component level. Perform developmental testing and documentation at the lowest level possible. Start at the component and subassembly level and design test programs to detect design weaknesses, defects, and thresholds where practical. Insufficient component testing causes many system-level failures. For a cost-effective test program, components must be tested to detect and eliminate failures related to poor quality. Worst case scenario, range of condition, limit testing, and interface testing should be done at the

component level to uncover potential issues as early as possible. Ensure that all design requirement verification and anomaly resolution is approved before the next level of integration.

- Establish conservative environmental test levels early in the program. As much as possible, the system should be tested in a "test-like-you-fly" configuration. A unit should not only be tested to send a command, it should be tested at the maximum required command rate. Redesign and/or retest associated with increased test requirements later in the program can cause major cost impacts, schedule delays, and operational failures.

- For problem resolution, base schedules on realistic and executable models that account for system production maturity, reasonable levels of integration returns, and realistic durations. Avoid success-oriented test programs that inevitably result in downstream problems (including system-level failures).

- Start implementation of security and standards requirements as soon as possible. Waiting until after a system is developed to address these could result in designing a system that needs to be reworked and retested due to being unable to satisfy mandatory standards.

- Inadequate test planning early in a program can cause significant cost and schedule impact. Major elements of a program can be affected by initial test plans, including schedules, facilities, test and handling equipment, training, safety, security, government-furnished services and equipment, transportation, and even design of the flight HW and SW.

- Plan for and implement a disciplined anomaly tracking and resolution process that determines root cause failure on all anomalies and includes all factory, subcontractor, and operational anomaly data. Ensure appropriate personnel are trained in the process. Ensure the root cause is determined and that corrective actions include all affected HW, SW, documentation, and organizations.

19.5 Government and Contractor Enabling Processes and Products

19.5.1 Required Resources

In order to support the test planning and verification processes, requirement specifications and interface documents must be available. Additionally, the

ground system architecture and subsystem interactions are needed. The SOW and any standards that must be followed, such as those related to security, safety, or human factors are required to ensure that the system is tested for compliance.

19.5.2 Documentation Products

Test plans, procedures, and the test documentation described below shall be collected and maintained in test documentation files. Developers shall maintain the test documentation files for the duration of development, operations, and maintenance.

Test Configuration Audit. For each test execution, a test configuration audit shall be performed to document the versions/builds and revision designators of all hardware, software, operating systems, configuration files and other items constituting the test environment. Test equipment used shall be listed with calibration dates and accuracy.

Test Data. Test data shall include the data required for preparation, simulation, or configuration, for performing the test, and data collected during and after the test. Test data shall be collected and maintained in a form to permit retest and the evaluation of performance under the various specified test conditions, and can be used to compare against future results to detect regressions.

Test Log. The results of formal tests shall be documented in a test log. The test log shall identify the personnel involved and be time-tagged to permit a reconstruction of test events such as start time, stop time, anomalies, procedure steps changed or not completed, and any periods of interruption.

Test Discrepancies. Anomalies, discrepancies, and failures occurring during test activities shall be documented and dispositioned as specified in the developer's quality control plan. Test discrepancy and resolution records shall be reported to the procuring activity as required in applicable contract or development agreement.

Qualification and Acceptance Test Report. For qualification and acceptance tests, test results shall be documented in test reports. The test report shall state the degree of success in meeting the test objectives and shall document the details and conclusions from the test results (including verification status of each requirement), and a summary of the test results, deficiencies, problems encountered, and problem resolutions.

19.6 References

1. Lutton, David, Colleen M. Ellis, James A. Shneer, Suellen Eslinger, and Brian E. Shaw. *Test Requirements for Ground Systems*, TR-2013-00215, The Aerospace Corporation, El Segundo, CA. 2013.

2. Englehart, William C. *Space Vehicle System Engineering Handbook*, TOR-2006(8506)-4494, The Aerospace Corporation, El Segundo, CA. 2005.

3. White, Julia D., Geoffry A. Larsen, and Dan W. Hanifen. *Space Vehicle Test and Evaluation Handbook*, TOR-2011(8591)-2, Volumes 1 and 2. The Aerospace Corporation, El Segundo, CA. 2012.

4. Adams, Richard J., Suellen Eslinger, Peter Hantos, Karen L. Owens, Linda T. Stephenson, Joanne M. Tagami, and Ronald B. Weiskopf. *Software Development Standard for Space Systems*, TOR-2004(3909)-3537B. The Aerospace Corporation, El Segundo, CA. 2005.

19.7 Acronyms

ADPE	automated data processing equipment
C&T	commanding and telemetry
CDR	critical design review
CDRL	contract deliverable requirements list
CM	configuration management
COTS	commercial off the shelf
CPU	central processing unit
DRE	discrepancy report
FCA	functional configuration audit
HW	hardware
I&T	integration and test
I/F	interfaces
IMP	integrated master plan
IPT	integrated product team
IPT	integrated product team
MTBF	mean time between failures
MTTR	mean time to recover
PCA	physical configuration audit
PDR	preliminary design review
RVTM	requirements verification traceability matrix
SDF	software development file
SDR	system design review
SOW	statement of work
SVR	system verification review
SW	software

TRR	test readiness review
VCRM	verification cross-reference matrix
WG	working group

Chapter 20
System Engineering Aspects of Test Like You Fly

Rachel D. Morford
Future and International Programs
MILSATCOM Division
Julia D. White
Systems Integration and Test Office
Mission Assurance Subdivision

20.1 Introduction/Background

The Test Like You Fly (TLYF) process is a pre-launch/pre-operational systems engineering process that provides a comprehensive approach to mission validation through translation of mission operations concepts into perceptive operability tests. For ground systems, the process could more accurately be called Test Like You Operate, with the same intent—use operationally realistic test scenarios to find potential flaws in the system, assuring the ability to perform its mission when it comes online.

The TLYF process defines steps to prevent mission flaws from escaping before a system becomes operational, with the goal of demonstrating that the system can successfully operate in realistic, mission-like conditions. TLYF is a collaborative, system-engineering process, beginning with an understanding of mission objectives and execution. TLYF implementation is comprised of both systems engineering activities, and test development activities as shown in Figure 20-1. These activities inform each other throughout the design and development of a program, ensuring an effective TLYF process. A tenet of the TYLF process demonstrates that you can fly/operate the mission, which is fundamentally different than demonstrating that the system meets requirements.

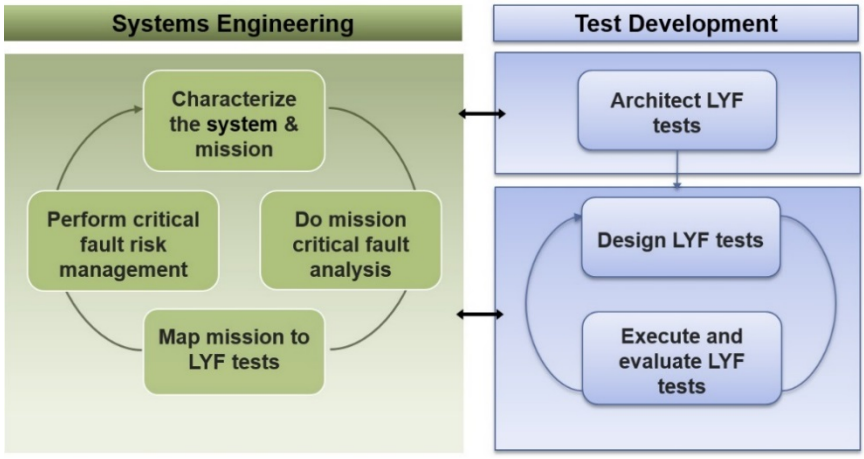


Figure 20-1. TLYF process.

20.2 The Test Like You Fly Process

TLYF is based on two principles. First, a system should never encounter an event for the first time while in operation. Second, a system should never be tested in a manner that could result in damage.

The TLYF system engineering activities, illustrated on the left-hand side of Figure 20-1, include characterizing the system and the mission, performing mission critical fault analysis (MCFA), mapping the mission to operationally realistic "Like You Fly" (LYF) tests, and performing critical fault risk management. These activities ensure that the operationally realistic tests capture essential mission characteristics and are perceptive to faults that can cause loss or severe degradation of the mission. Once data has been received from the systems engineering activities, test architecture and design can begin. The system and mission information provides the foundation for architecting LYF tests. Test development activities, shown on the right-hand side of Figure 20-1, include mapping the mission to LYF tests, architecting LYF tests, designing LYF tests, and executing and evaluating LYF tests.

Table 20-1 captures a summary of the steps in the TYLF process. A full description of the process is found in Aerospace report TOR-2014-02537, "The Test Like You Fly Process Guide for Space, Launch, and Ground Systems" [1].

Table 20-1. TLYF Process Implementation Overview by Steps

TLYF Steps	General Description
Step 1: Characterize the System and Mission	Before anyone can TLYF, it is necessary to know the system and its mission. This part of the process is centered on answers to a series of questions. System Aspects: The starting point for a TLYF assessment is understanding the "What" and "How" of the mission (i.e., what elements are involved and how they interface, and interact, internally and externally). It is important to understand the basic system elements, structure, architecture, interfaces, and design. How should it work? Mission Aspects: The mission aspects reveal the "How" and "When" of system operation, what defines mission success, and the details of its resulting services or products. The concept of mission-critical activities becomes a foundation for the later subject of fault analysis of mission-critical situations.
Step 2: Do Mission Critical Fault Analysis	MCFA is a top-down analysis of potential failures that is performed before a system is fielded or flown. The MCFA starts with identifying the modes of unrecoverable mission failure. It is important to note that mission failure is not restricted to a spacecraft or payload failure. It can be something that disrupts operations to the degree that the mission cannot be successfully executed. For each mission failure mode, the analysis identifies the root cause(s) of the failure; for each root cause(s), there should be an exoneration path that demonstrates that the potential fault is not present in the system. Executing a thorough MCFA requires a team, led by a systems engineer that includes experts from each system and subsystem.
Step 3: Map Mission to LYF Tests	Candidate tests are identified and proposed from the mission concepts, phases, and timelines. Test objectives are tied closely to mission objectives. There is special emphasis on first-time and mission-critical events in all mission phases. Candidate LYF tests may also be derived from the mission critical fault analysis, particularly to surface flaws that are associated with lower levels of integration or particular mission scenarios. Allocated LYF tests result from this step.

TLYF Steps	General Description
Step 4: Architect LYF Tests	Allocated tests must be architected (who, what, where, when, why, and how), designed, and executed. Architecture and design will involve trade-offs among objectives, the mission characteristics incorporated, and fidelity to the way the activities will be executed during the mission. The architect must be responsible for resource management and initial identification of LYF test exceptions.
Step 5: Design LYF Tests	LYF tests design involves selecting test cases needed to meet the test objectives; selecting which mission characteristics need to be included and over what range; deciding on initial and end conditions; and determining what test equipment and simulations will be necessary to execute the test. The test designer is responsible for identifying and evaluating LYF test exceptions.
Step 6: Execute and Evaluate LYF Tests	The execution and evaluation of LYF tests should provide evidence for operability, exonerated flaws from MCFA, and detect flaws preventing mission success. Tests that identify flaws will provide the basis for additional tests (regression or new). Evaluation of LYF tests includes both the degree to which the test results meet the success criteria and the necessity to re-run the test following changes to hardware, software, processes, or procedures.
Step 7: Critical Fault Risk Management	Critical fault risk management encompasses identification, analysis, mitigation planning and implementation, monitoring, and elevation of critical fault (CF)-related risks. The mission critical risks identified are based on: (1) the potential flaw paths to mission critical failure situations, as an output of the MCFA, and (2) the exceptions identified during the LYF test design process. The assessment process accounts for the risk of not being able to test in a flight-like manner by evaluating LYF test exceptions. High-risk items remaining as a result of the critical fault risk management assessment become candidates for program risk management decisions. Management may allocate additional resources to mitigate the risks stemming from LYF exceptions.

The remainder of this chapter concentrates on aspects specific, to or particularly significant in, operationally realistic testing of ground systems.

20.3 Ground Systems Special Considerations for TLYF

The TLYF process can be applied to all aspects of ground systems, including—but not limited to—command and control centers, data processing centers, payload processing facilities at launch sites, backup command centers at a developer site, a set of workstations in a maritime patrol plane, or a hand-carried computer in a combat zone. It is essential to include operators, users, and other stakeholders when evaluating and implementing the TLYF process. When considering the mission characteristics for ground systems in the TLYF process, it is important to recognize that they may be broader in scope than for space assets and that they include characteristics of the space assets that they support. The use of personnel in all processes—including those related to efficiency, health, and safety—needs to be included in TLYF process planning.

In planning for ground system testing, especially at the segment and system level, test engineers need to be perceptive regarding the environment in which the testing takes place. Many interfaces not owned by the contractor need to be engaged, with the majority of those interfaces outside the unit test or dry-run boundaries. The more simulated the interfaces, the higher the risk (although the greater the control). Defining environments needed for each test cannot always be specified during early test planning.

Table 20-2 lists specific characteristics that should be considered in the TYLF process. In test execution, many of these test elements can be detractors to successful mission completion, if not contributors to failure.

Table 20-2. Mission and Test Characteristics

Test Elements	Specific Test Areas	Test Perceptiveness (Boundaries, Data, Configurations)
End-to-end (integration) level	User terminals, mission control center, backup mission control center, payload processing center, tasking authorities, communication services, space vehicles (SV)	Ground operations equipment, ground operations software
End-to-end (integration) level	User terminals, mission control centers, backup mission control center, payload processing center, tasking authorities, communication services, SVs and payloads (PLs), end users	Ground operations equipment, ground operations, operators, facilities, maintenance, software

Test Elements	Specific Test Areas	Test Perceptiveness (Boundaries, Data, Configurations)
Time and timeline	Running clock, preprogrammed/invariant command sequence, fixed duration activity, variable duration activity, order dependent activity, order independent activity, initial conditions, first time activity, simultaneous activities	Clock reset, test driven function order, test initial conditions, test duration, test recurrence rate
Environments (internal, ascent, space)	Location, geography, neighborhood, weather and climate, facilities, power, cooling, system loading	Operator workstations, simulators, room layout
Configuration	SV space assets, ground SV control (uplink), ground PL control (uplink), ground and interfaces	Non-flight test article, test equipment, test software, facility, simulators, test support functions
Telemetry (state of health, mission data, payload data)	bands/rates, automated limit telemetry checking, stored telemetry playback, manual/ real-time telemetry (TLM) evaluation, TLM trending and evaluation (tools), mission data production tools, databases, data type(s)	Telemetry hardline, special test equipment (STE) limit checking, test TLM database. data formats, TLM processing, payload models, simulated payload data
Time and timeline	Running clock, sim clock, preprogrammed/invariant command sequence, fixed duration activity, variable duration activity, order dependent activity, order independent activity, initial conditions, first time activity, simultaneous activities	Clock reset, test driven function order, test initial conditions, test duration, test recurrence rate
Environments	Location, geography, neighborhood, climate (weather), power, cooling	Operator workstations, noise and communication interference, room layout
Infrastructure (facilities and comm)	Location, facilities, power, cooling, secure voice, secure data, bandwidth, cyber, physical security protection	Quality of data, timeliness of data, data availability, evaluation of test data

Test Elements	Specific Test Areas	Test Perceptiveness (Boundaries, Data, Configurations)
Uplink (commands)	Bands/rates, contact plans, command plans, time-tagged commanding (CMD uploads), real-time commanding, command options, automated fault management	Command hard line, test command script, test command database, autonomous commanding, ground antennae locations and availability, SV in-view times, SV simulators, SV cryptos
Telemetry (downlink)	Configurations (bands, formats, stored/real-time, rates), automated limit telemetry checking, stored telemetry playback, manual/real-time TLM evaluation, TLM trending and evaluation (tools), mission data production tools, databases and data type(s)	Telemetry hardline, telemetry paths, commanding paths, STE limit checking, test TLM database, data formats, TLM processing, payload models, simulated payload data, data evaluation/trending/production, SV cryptos
Mission planning	Payload planning, mission planning, mission procedures, mission processes, mission phases, operational modes, crew training	Test procedures, test processes, test conductors, training simulators, external interfaces
People	Training, certifications	Communications, certifications, length of time on program, cohesiveness, ability/knowledge

20.3.1 Mission Capability Growth

The TLYF process is a mission assurance technique that should be applied to new systems, established systems being upgraded with new technologies within current capabilities, and the addition of new capabilities to existing systems. It is this addition of new capabilities that represents a key difference in how the TLYF process is applied to ground systems versus other components of the space enterprise. Builds, increments, effectivities, and deliveries are terms that are used to indicate the growth of a ground system's capability. This capability growth involves changes to hardware, software, operational procedures, mission processes, and personnel utilization. The capability change may also include changes to facilities to support the activity of the system, the environmental conditions, and external users and services. "Environmental conditions" may refer to the operational environment within a command and control ground system, or may refer to external and internal physical environments (ocean exposure, ambient pressure and temperature, winds, etc.). A ground system is

likely to have several cycles of capability growth over its full lifecycle. TLYF processes need to be in place and geared to both short-term development and long-term maintenance activities.

20.3.2 First Time and Mission Critical Events

In applying the TLYF process to satellite and launch systems, much of the focus is on first-time, mission-critical events in order to comply with the fundamental philosophy established by Shelton and Roskie's seminal paper on this subject [2]: a vehicle should not experience either mission critical operations, environments or stresses for the first time during flight. For ground systems, the principle still holds, although the identification of "first time" operational events may not be tied to a single point in time.

First-time and mission-critical events might not be as obvious in a ground system as in a space system. For instance, the first time a new function in the software is used is a first-time event, although it is likely to be part of a larger event such as commanding a new mode in a space asset. Sometimes it will be more subtle, such as when a new software delivery is tested. In this case, the first use of the capability may be during a ground dry run against an existing space asset.

Ground system acquirers and developers should look carefully at product and process introductions to adequately determine an appropriate set of first-time events for consideration in operationally realistic testing. External systems that could affect the operations of the system under test should be operated or simulated to replicate conditions expected during operations. Testing should include a focus on external interfaces, the use of operational databases, operational scenarios, and system requirements from a mission operations perspective.

One way to pull many first-time ground/space activities to earlier in the development cycle is to acquire the ground system in tandem with the space system and use the ground system command and control functionality during factory space systems tests. This is a best practice identified by APL [3], JPL [4], and AFRL at the Test Like You Fly session at the 2008 Ground System Architectures Workshop. This workshop emphasized the applicability of the TLYF process to people, organizations, and mission processes (planning, commanding, data evaluation). This approach ensures that most "first-time" usage of new software, new hardware, new procedures, patched software, repaired hardware, and revised procedures all occur well ahead of the actual mission.

20.3.3 System Upgrades

Ground systems require sustainment for many years, even decades. A new ground system acquisition should cover initial usage, planned additional capabilities (to be included in subsequent deliveries), and strategies for longer-term sustainment. All of these factors need consideration when applying TYLF processes to ground systems. Each new aspect or upgrade of the ground system should be re-assessed from the TLYF process perspective. Does the new aspect affect mission processes, timing, and transactions? Is there a change to the way a test is run to ensure that it remains mission-like? Does the change add the possibility of a new failure mode? These and similar questions need to be asked and answered to ensure the new aspects of the system don't have severe ramifications.

20.3.4 Models and Simulations

All LYF tests rely on models and simulations to represent aspects of the mission that are not directly available during test. While it is important to recognize that exactly WHAT is operated/flown needs to be tested, there is always going to be a need to model or simulate those things that can't be included in the test. Appropriate models and simulations will help characterize the complexity of ground system architectures and operations, and they will provide a range of performance characteristics and operational parameters. Candidates for models or simulations include process times, transaction latency, processor utilization, and queue memory usage. Development and validation of these models and simulations can be a significant part of a successful system development effort. As part of the development of these tools, it is necessary to track the deviations between the real world and the simulation; these deviations are the basis for the LYF test exceptions.

The operational realism of the LYF tests is dependent on the similarity of the simulations used in testing to their operational analogs. It is necessary to verify and validate the algorithms and the simulations against actual systems. For ground systems, the models can help determine the expected workload and pacing of operations. For example, an external tasking or certification authority may be used that drives the operations of the system. Visibility into the behavior of these external systems may be limited and require early coordination with other organizations in order to have the simulations ready for testing.

20.3.5 Automated Internal Monitoring

Ground systems often have critical chokepoints that are not readily visible or accessible to system operators. These parameters can include the available space on storage devices, inter-process queues, and the utilization of network bandwidth. Often, such parameters have requirements defining thresholds, but

the systems need specialized test equipment or analysis to verify the activity levels during tests that replicate mission conditions.

One solution that has shown great benefit is the inclusion of automated monitors throughout the system that report to a central system management function. Several protocols exist for monitoring distributed systems that can also be used to report the status of internal software conditions, such as CPU utilization, queue lengths, process latency, and transactions per second. Not only does this make testing easier, in enables the operators to detect problems before they affect the mission. That said, if these monitors are not included as part of the operational system, they in turn become LYF test exceptions.

20.3.6 Human Factors

When characterizing a ground system, it is important to account for human factors, as operators need to be considered as a part of the system. For the purposes of TLYF, we are mostly concerned with the roles and responsibilities, expertise, schedules, tasks, and workloads of the operators and/or users.

For ground systems, including manning plans and schedules, roles and responsibilities of operators and users, training and hiring plans, and the human element of facility, hardware and software maintenance plans should be included when evaluating operationally realistic testing.

Additionally, because ground systems include personnel and supporting facilities, the variety of failure modes is broader. These failure modes should be tested and accounted for in operationally realistic testing. Operator errors are a source of potential failure modes; these failures should be simulated based on lessons learned during testing. The ground facilities themselves introduce failure modes such as fires, leaking pipes, and power outages from defective office equipment. In addition, ground systems provide more access points for cyber attacks, both from the network-access perspective and from the physical security of the data.

20.3.7 Availability and Repairs

Ground systems are perceived as easier to repair than space assets because they are accessible and generally any component can be replaced. However, building a system that can be repaired during operations is not simple. Availability requirements place constraints on maintenance and repairs and the implied level of reparability of the system implies some requirement for diagnostic capability. Maintenance and repairs during operations need to be considered in the TLYF process and in operationally realistic testing, as do the diagnostic capabilities of the system.

Ground systems require planned/preventative and unplanned/corrective maintenance, and component replacements. This is easier and less of an operational impact if the system has multiple strings of equipment and several cross-over points to configure the operational path. Testing of upgrades is much easier if part of the system can be segregated into a test string. Ideally, a system should be able to split into two complete systems with one operational and the other in shadow mode using the updated component, which allows the ground system to run operational tests without interfering with the mission. Mobile ground systems may not have this luxury, but reparability and availability are still issues that need to be addressed and also tested on mobile units.

20.3.8 Distributed Systems

Another factor to consider in a ground system is that it might be distributed over a large geographic area with multiple and diverse communication links. Some communication lines may be outside the control of the system, such as the internet, and may be difficult to include in operationally realistic testing. Synchronization and timing of events may have a significant impact on mission capabilities. Emulation of these external assets may be needed and access to the actual assets may require advanced planning to prepare for testing. Distributed systems are also subject to cybersecurity considerations. TLYF should be considered when planning cybersecurity tests.

20.3.9 Characterizing External Entities

For ground systems, what we "fly" often includes external services, organizations, and equipment, which are not part of the system but are critical to its function. These may be communication services, tasking or planning authorities, or radio frequency (RF) equipment, among others. Defining and characterizing them is necessary to allow proper test planning and allocation of resources.

20.4 Examples of Ground Systems Lessons Learned

The Test Like You Fly process evolved from several mission failures or near-failures and the lessons learned from those investigations. Among the failures that impacted the development of Test Like You Fly were ground segment failures. Two of those are described here, with an emphasis on understanding how operationally realistic testing may have impacted the mission to have a different outcome.

20.4.1 Mars Climate Orbiter (MCO) Crash

Launched on December 11, 1998, the Mars Climate Orbiter (MCO) broke up near Mars. The failure was determined to be due to a mix-up between English

and metric units during ground software development, coupled with an insufficient navigational approach.

The MCO spacecraft was built by Lockheed-Martin for the Jet Propulsion Laboratory (JPL) under the "Faster, Better, Cheaper" banner. It derived considerable heritage from previous Mars missions, especially the successful Mars Global Surveyor (MGS). MCO was equipped with thrusters used for angular momentum damping and trajectory corrections en route to Mars.

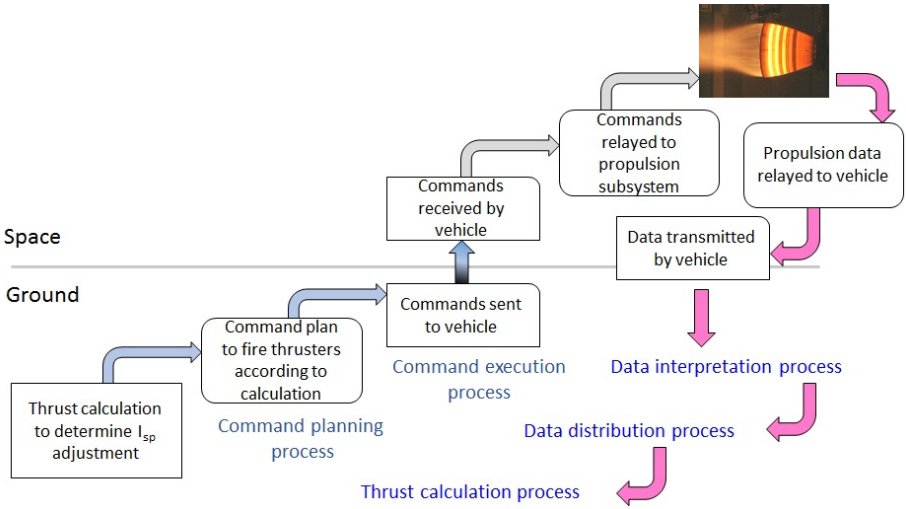


Figure 20-2. MCO orbit determination process.

For MCO orbit determination, all accelerations encountered during cruise had to be accounted for; however, it was not possible to directly measure each of the small accelerations. Instead, each thruster phase and duration was converted by "small_forces" ground software (see Figure 20-2), the same approach that had been used for MGS. The calculation software used in this ground software was also a reuse of MGS software. The original MGS software specification set forth that the ground software output should be in Newton-sec., a requirement that MCO followed. A change in thruster hardware made it necessary to update the model.

MGS and MCO used the same vendor for the thruster hardware. This vendor provided the propulsion equations in English units (lb.-force-sec.) each time. For the MGS model, Lockheed Martin engineers correctly added the 4.45 conversion factor to the vendor's equation. Unfortunately, the MCO team simply made a substitution of the English units for metric units. There was no warning in the ground software code comments and the MCO team overlooked the interface specification.

The ground software was labeled as non-mission critical, so it was not rigorously reviewed; the "truth" table, computed manually for acceptance testing, had the same units error. The interface of the "small_forces" ground software with the navigation function was only informally tested. This testing was not to verify the file, but to ensure that it could move between servers.

Ultimately, the probe was discovered to be over a hundred kilometers off course; this was unfortunately not realized until aero braking, after Martian gravity had captured the spacecraft and the true position of MCO could be calculated.

Test Like You Fly lessons learned from the MCO experience were:

- Rigorous formal testing is essential to validate changes in mission-critical software
- Expected results used in verification tests should be generated independently in accordance with system requirements
- Systems engineering processes must track giver/receiver transactions to highlight connections to critical items and activities.
- Proper identification of LYF test exceptions to point to areas not covered by tests.

20.4.2 Sea Launch F3 Failure

The third Sea Launch failed to place its satellite into a medium Earth orbit on March 12, 2000. The failure occurred because a conditional statement was omitted from a line of ground software code, thereby preventing a second stage valve from closing. The error was not caught because software changes were not rigorously controlled and because the mistake was not noticed during ground rehearsals.

Sea Launch, a consortium set up in 1995, uses a mobile platform to launch commercial satellites on a Zenit rocket. The second stage pneumatic system uses six helium bottles to control valve actuation, engine gimballing, and the spin start of the second stage engine. The bottles are isolated from the helium line with closed isolation valves that are opened with high pressure nitrogen supplied from the platform via valve VZT23 (see Figure 20-2). This valve was supposed to close at T-105 seconds by ground command.

This satellite launch required the launch countdown to start at a precise time. Accordingly, time in the ground software changed from a relative variable, "REFERENCE_TIME," to an absolute variable, "FIXED_TIME."

The mission timeline logic for the closure of valve VZT23 was originally:

- If the state is ABORT (or if the state is NOMINAL_LAUNCH and REFERENCE_ TIME is T-105 seconds), then close VZT23.

It should have been changed to:

- If the state is ABORT (or if the state is NOMINAL_LAUNCH and FIXED_TIME is T-105 seconds), then close VZT23.

Instead, it was changed to:

- If the state is ABORT, then close VZT23.

This line of code, as written, would cause the valve to remain open in a nominal countdown.

The change notice for this code change included several unrelated items, did not provide an explanation as to why the control code was changed, and did not compare the "was"/"is" algorithms. Testing was incomplete—not all manifestations (including displays) of success paths were checked.

Although the launch was rehearsed three times, console operators missed the open valve amid the ten thousand parameters that they monitored simultaneously. The rehearsals were not considered software tests, so there was no automatic configuration monitoring.

During first stage flight, helium leaked out through the fire suppressant system, leaving inadequate pressure to open the fuel valve of the second stage engine. The rocket lost control and the satellite was destroyed.

Test Like You Fly lessons learned include:

- Operational rehearsals are not LYF tests. However, they can be an opportunity to identify human factors influences on a system.
- Changes/updates to software should undergo the same TLYF rigor as initial software builds.
- Data transactions should be well understood across interfaces and system boundaries.

20.5 Summary

Test Like You Fly, or Test Like You Operate in the case of ground systems, is a systems engineering process that helps to ensure a system will be operational from day one. Like You Fly tests, defined to be perceptive through a clear

understanding of the system and the mission, are conducted before the system comes online. This level of analysis and testing helps to uncover risks based on a mission critical perspective of flaws and flaw paths. In the operationally realistic, system-level testing of ground systems it is vital to test the fully integrated system and to demonstrate that each of the individual elements can operate together to support the intended mission.

20.6 References

1. White, J. D., Tilney, L. G., The Test Like You Fly Process Guide for Space, Launch, and Ground Systems, TOR-2014-02537. The Aerospace Corporation, El Segundo, CA. 2014. Distribution limited

2. D. L. Shelton and S. C. Roskie, "Applying the Test Like You Fly Principle", in Proceedings of 19th Aerospace Testing Seminar, The Aerospace Corporation, El Segundo, CA. 2000.

3. Walter L. Mitnick, "Test Like You Fly" at APL, Ground System Architectures Workshop, April 2, 2008.

4. Ben Jai and Robin O'Brien, "Test Like You Fly (TLYF) Philosophy Applied to Ground Segment Testing," Ground System Architectures Workshop, April 2, 2008.

20.7 Bibliography

White, J. D., and L. G. Tilney. "Introduction to the Test-Like-You-Fly Process, Parts 1 & 2." Briefing presented at the 26th Aerospace Testing Seminar. The Aerospace Corporation, El Segundo, CA. March 2011.

ECSS E-10-03A, *Space Engineering Testing*, February 2002.

White, J. D., and Charles Wright. *End-to-End Testing in a Test Like You Fly Context*. presented at the 23rd Aerospace Testing Seminar. The Aerospace Corporation, El Segundo, CA. October 2006.

Arnheim, B., and Wright, C. *Insight into the Effectiveness of System Level Thermal Vacuum Testing*. Briefing presented at the 21st Aerospace Testing Seminar. USAF/The Aerospace Corporation, El Segundo, CA. March 2003.

SMC Standard SMC-S-012. 13 June 2008.

Beutelschies, Guy. That One's Gotta Work. Mars Odyssey's Use of a Fault Tree Driven Risk Assessment Process. Aerospace Conference Proceedings. IEEE. 2002.

Hogan, Steve. *Effective Fault Management Guidelines*. TOR-2009(8591)-14. The Aerospace Corporation, El Segundo, CA. 2009. Distribution limited.

White, J. D., and C. Wright. "Test Like You Fly: A Risk Management Approach." *Space Systems Engineering and Risk Management Symposium*. The Aerospace Corporation, El Segundo, CA. October 2005.

Metodi, Tzvetan S. *Space Vehicle Testbeds and Simulators Taxonomy and Development Guide*. TOR-2010(8591)-16. The Aerospace Corporation, El Segundo, CA. June 2010. Distribution limited.

White, J. D., Tilney, L. G., and Knight F. L. *Test Like You Fly: Assessment and Implementation Process*. TOR-2010(8591)-6. The Aerospace Corporation, El Segundo, CA. January 2010.

Bergen, T. L., and White J. D. *Guideline for Performing Test-Like-you-Fly Assessments*. TOR-2011(1315)-1. The Aerospace Corporation, El Segundo, CA.

20.8 Acronyms

AFRL	Air Force Research Laboratory
APL	applied physics laboratory
CMD	command
CPU	central processing unit
ISP	specific impulse
JPL	Jet Propulsion Laboratory
LYF	like you fly
MCFA	mission critical fault analysis
MCO	Mars Climate Orbiter
MEO	medium Earth orbit
MGS	Mars Global Surveyor
PL	payload
RF	radio frequency
STE	special test equipment
SV	space vehicle
TLM	telemetry
TYLF	test like you fly

Chapter 21
Transition to Operations

Wayne G. Yenne
Computer Applications and Assurance Subdivision
Computers and Software Division

21.1 Introduction/Background

This chapter will describe the process by which a ground system is transitioned from development to operations in the field, the theory and principles behind successfully accomplishing this transition, and the mission assurance approach used to assess the readiness of a system to be transitioned and used operationally.

Transition to operations is the final step of the "organize, train, and equip" effort. For some systems, particularly those that are replacing a legacy system supporting an operational satellite constellation, transition is complicated by the system's operational status. Transition planning is critical to ensure that impact to the legacy system's operational mission is minimized during the transition.

Successful transition begins with careful planning from the start of the program and is emphasized through the development process. A failed transition, or in other words a system that fails to achieve a formal operational acceptance decision after entry into operational test, can have serious consequences on a program. Those consequences can include cost overruns, delays in fulfilling operational mission requirements, and degradation of end user capability. An independent evaluation of readiness of a system to be transitioned to operations should be conducted by using the assessment areas laid out in this chapter.

21.2 Definitions

Developmental test (DT) Any testing performed by the contractor or government acquisition agency prior to the operational test.

Operational acceptance (OA) The final step of the transition. This occurs after a successful operational test and all the established operational acceptance criteria have been met. Operational acceptance is the milestone that marks the transition of the system from the developer to the operator. The system may have been in operations prior to this declaration under a trial period.

Operational readiness The readiness of a system to enter operational test and subsequently obtain operational acceptance. The terms operational readiness and transition (or system transition) are used somewhat interchangeably.

Operational test (OT) Testing performed by an independent test agency [e.g., Air Force Operational Test and Evaluations Center (AFOTEC)] in the operational environment with operations personnel to determine if the system meets its mission requirements and is suitable for operations. Generally, this test is the final step before OA.

Transition (or system transition) The movement of a system from one state to another. In this context, the term is used to describe the entire process (planning and execution) required to move the system from development into the operational environment. The term encompasses development activities as well as the operational test and operational acceptance processes.

Trial period An interim operational period conducted under ground rules agreed to by the developer and operator.

21.3 Transition to Operations Process

Figure 21-1 shows the final milestones of a system transitioning from development to operations for an Air Force ground segment. Other acquisition organizations have similar milestone assessment points. Before a program gets to these milestones, the program has been planning for its transition to operations in conjunction with the development of the system.

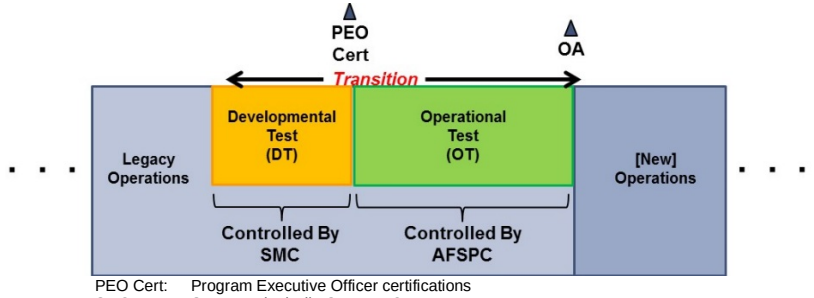


Figure 21-1. Transition milestone from development to operations.

A developed system must pass an operational test in order to be accepted for operational use. Before the OT can be conducted the SMC Commander (SMC/CC) certifies the system is ready. This is known as a Program Executive Officer (PEO) Certification. This certification is based on checklists contained in Air Force Manual (AFMAN) 63-119 [1]. When a system is used operationally prior to obtaining operational acceptance, it is said to be in a trial period. AFSPC approves a system entering trial period based on a set of criteria created with the development and operations stakeholders. Depending on the system, the trial

period may be entered to finish development test, start operational test, or at the completion of operational test.

The government program office (to include support from The Aerospace Corporation) has a critical role in advising senior leadership regarding the readiness of a program to enter trial periods and operational test. The program office provides input to the SMC Commander for the PEO Certification via two distinct and separate paths. First, SMC/EN, the engineering arm of SMC, provides a recommendation to SMC/CC based on the AFMAN 63-119 checklists. Program office engineers provide SMC/EN customers with aspects of that assessment. Second, Aerospace Corporation engineers perform independent assessments of readiness, and provide those assessment to SMC via the Aerospace Corporation President or Vice Presidents. The Aerospace President's Review (APR) (or similar review is held to communicate the assessments from the program office to The Aerospace Corporation executives). Other acquisition organizations have similar key milestones which include acceptance at senior levels.

21.4 Summary of Tasks/Principles

This section provides the theory behind transitions by listing the groups of products to be considered in planning for transition and enumerating the principles of a successful transition.

21.4.1 Transition Product Groups

The elements to consider in transitioning a system to operations can be binned into the five product groups below. These product groups break the system down into sub-groups of like elements to facilitate understanding. All five groups of these must be evaluated for transition readiness.

(a) **Physical System—**The physical system consists of all the elements that contribute to the performance of the mission. For a satellite ground control system, it would include the mission control facility, organic remote sites, and external interfaces.

(b) **System Support**—System support elements are those items that enable the effective and efficient operation, maintenance, and sustainment of the system. This includes, but is not limited to: the system spares, technical documentation, operations and maintenance procedures, sustainment capability, training documentation, and training systems.

(c) **Certification and Assessment Documentation**—Certification documentation that proves the system meets requirements or supports the testing or fielding decisions. These transition artifacts include but

are not limited to: Functional Configuration Audit (FCA)/Physical Configuration Audit (PCA), AFMAN 63-119 Certification (or equivalent), Information Assurance (IA) certifications, and operational test reports. Additionally, programs often have some number of "independent reviews" conducted by an independent readiness review team (IRRT), independent contractor team, or other "outside" entity. The artifacts and findings from all these reviews may be contradictory.

(d) **System Deployment Tools, Processes and Procedures**—This group consists of any mission unique tools, processes, and procedures needed to enable the system deployment. Often with existing space capabilities, major segments are fielded to replace or upgrade the system while the system is in operations. Thus, the system is required to operate through the upgrade. In order to field while operating, special tools, processes and procedures often need to be developed to facilitate the transition.

(e) **Personnel and Resources**—While not a developed product, the availability and readiness of the people responsible for operating and supporting the system is critical for a transition. Personnel need to be assigned and trained to support development, operational readiness, and operational testing of the system and to perform and sustain the mission of the system once placed into operations.

21.4.2 Transition Principles

Underlying the mission assurance approach to operational readiness is a set of transition principles. These principles derive from extensive Aerospace program transition experience, incorporating best practices and lessons learned [2]. These principles illustrate the importance of baking-in transition planning from the earliest phases of the program to ensure success. Although transition or fielding is often viewed as the last phase, or a separate stand-alone phase, of the program, the mission assurance approach for operational readiness assesses how a program is applying these principles to the development and execution of the transition from the beginning of the program through all the major transition or fielding events.

(a) **Early Engagement**—As with any effort, one needs to start with the end result in mind. For transition, consideration of how the system will be transitioned to operations needs to be included in the material solutions analysis phase (pre-Milestone A). This early engagement reveals requirements or considerations that will establish the conditions for transition success.

(b) **Plan and Program for Transition**—Planning and programming for transition activities are critical to transition success. Proper planning and programming ensures adequate resources (funding, personnel, time, contract mechanisms, etc.) for transition execution. Additionally, comprehensive, deliberate planning allows for better resource usage and coordination, and ensures products required to certify the system are collected as they are developed.

(c) **Proper Emphasis on Transition**—Transition planning activities evaluate a number of factors (type of deployment, system complexity, transition complexity, user impact, etc.) and their implications to the program. Care must be taken to ensure system transition activities are properly emphasized and managed at a level commensurate with these implications. In cases where there are significant implications associated with the system transition to operations, establishment of a separate organizational element tasked with ensuring the safe and timely transition of the system may be warranted.

(d) **Expectation Management**—Programs should ensure expectations among stakeholder organizations are identified and managed through the program life. Successful expectation management reduces the number of significant issues and surprises among the stakeholders, which include the acquisition, sustainment, and operational user communities. Some programs have used formal expectation management agreements to help define, resolve, or ensure agreement of potential expectation differences in the areas of cost, schedule, and performance.

(e) **High Fidelity, Robust Testing**—High fidelity, robust testing is a critical principle of transition as it demonstrates that the developed system not only meets its formal requirements, but more importantly meets the operational need and supports key acquisition end-game decisions (e.g., readiness for OT, trial period entry, operational acceptance, etc.). This is accomplished through a systematic, disciplined testing and evaluation program that includes development testing (verify critical technical parameters) and operational testing (verify capability-based requirements). The testing program is critical to building confidence in the system's ability to perform the operational mission. This includes testing of the end-to-end system in its operational configuration and environment prior to delivery of the system to the government. In addition, the testing should include as many of the system's elements (mission control centers, remote sites, spacecraft, user/external interfaces, etc.) as possible to reduce operational risk.

To get to this level requires multiple layers of verification of increasing fidelity and robustness. An example of testing progression would move through the component, element, segments, systems, and enterprise levels. It would also move from software simulation, hardware simulation, off-line testing with operational assets, and live testing with operational assets (if operational assets are available).

21.5 Practices—Operational Readiness Mission Assurance Tasks

The assessment tasks from the Mission Assurance Baseline used for operational (or transition) readiness are listed and described as they capture best practices and constitute the Aerospace recommendations to maximize the probability of successful transitions to operations [3].

21.5.1 Physical System and System Support Readiness

Physical system in this case is defined as the system to be fielded. This task contains assessment activities that address the readiness and maturity of the developed system to be transitioned into operations. The fundamental question to be addressed by the assessor is: "Is the developed system ready to be fielded?"

Task	Description
Test Procedures are Operationally Realistic	There is a risk in a requirements-based development test program that the product passes test, but is not operationally suitable. Tests procedures should be looked at from the point of view of the ultimate operators and the operational environment in addition to a requirements fulfillment point of view. They should include test-like-you-fly (TLYF) and day-in-the-life concepts and ensure testing is consistent with the intended operational use of the system. Using actual operators to use the system in an operationally realistic multi-day scenario is an example of this principle. Situations where test configuration limitations create transition and operational risk need to be highlighted and mitigated when possible.
Support Equipment Used for Developmental Test Is Sufficient for Determining Operational Readiness	Nothing can replace testing in the operational environment, but this is usually not feasible for space systems. This task asks the reviewer to ensure simulation software and hardware are validated, and are an adequate representation of the operational environment. Simulation tools should be validated, and an assessment made to determine if those tools can exercise maximum and minimum loading conditions, non-nominal conditions, error recovery, bus and payload anomalies, etc. Where possible, ensure actual operational data are used in the test environment.
Ensure Stability and Stress Tests are Provided in the Developmental Test Plan and Procedures, and that	Every possible operational configuration should be tested as a part of the developmental test plan. This includes both ground and space segment assets, and voice/non-space communications links. Test cases should exercise all portions of code—off-nominal

Task	Description
those Tests Cover Off-Nominal Cases, Backup Locations and Data Paths, and Worst-Case Scenarios	cases, error conditions, all operator or administrator responses, and so on. Stability and stress test (network loading, operational interfaces, off-nominal conditions) procedures should be used to exercise maximum data volume, maximum operator interface, etc. Testing must be conducted for long durations without breaks to demonstrate stability under operational load (e.g., TLYF, day in the life, week in the life etc.).
Operational Databases, Whether Real-Time or Off-Line, are Complete and Sufficient for Operational Test	Testing should be done with operationally realistic data populated in the system's databases. The pedigree of the data used in test should be known. Test plans should include flows of the operational data through all paths, including offline and analysis software.
Ensure the Developmental Test Plans and Procedures Cover Off-Nominal Use Conditions and Threads	Ensure developmental testing covers as many off-nominal threads as possible. Problems typically don't occur with the technical issues, like bits in and bits out or performing nominal tasks. Rather problems occur while performing seldom used tasks, while being used in a fashion not expected by the developers or while trying to recover from other problems. Pressing the enter key repeatedly and making attempts at shutting down processes during loads are examples of off-nominal tests. Tests of nominal cases should always have success. There are different ways to define the off-nominal conditions, but there should be consensus with the operational community about what constitutes "realistic" off-nominal scenarios.
Ensure Maximum use of Real-Time Equipment in all Aspects of Developmental Test	Live data, hardware, and other operational assets should be used for developmental test whenever possible. If simulated data inputs are used, make sure they are sufficient to verify test results.
Ensure Operators are Continuously Involved with Product Development, and that Feedback from Operators has been Captured, and Adjudicated by Both the Provider and Receiver of the Products	Operational stakeholders need to have input into the design and use of the system to be fielded from the beginning of the program. Examples are conducting focus groups with operators when designing human-machine interfaces (HMI) and using operators during the testing. Comments from operators should be gathered and adjudicated whenever they are involved in developmental activities. If operators are used during testing, ensure they have received training commensurate with their test duties.
Ensure Operational Test Objectives are Reflected in the Developmental Test Program	If the development test program does not reflect the planned operational testing, there is a risk that operational test will uncover problems very late in the program and delay operational acceptance. Relevant operational test cases need to be reflected in the developmental test program. The developmental test (DT) program should review OT plans before the OT plans are approved to ensure the OT program will not test something not covered in DT .
Ensure Involvement of Operational Testers in Developmental Test	The developmental test staffing should include operational test team members in some capacity. This provides invaluable insight to operations, which can help avoid surprises at entry to operational test.
Ensure Concepts of Operations (with which the	Often, given lengthy development cycles, the concept of operations that influenced the initial design of the system has

Task	Description
Contractor Developed the System) are not changed Without Commensurate Operations Involvement, Test Plan and Procedure Changes, and Requirements Changes	changed significantly by the time the system is to be fielded. The organizations involved may have changed. The expectation of the skill levels of the operators may have changed. An effective transition program will monitor any changes to the concept of operations and ensure that impacts to system development are evaluated and adjudicated.
Ensure the Operational Units are Involved in Development of Operations Concepts, Implementation Reviews, and Developmental Test Activities	Operational units need to be engaged with contractor/government test personnel so that operational implication of problems discovered in test are known. In ensuring consistency with the operational concepts to be employed, operators should be involved before and during developmental test.
Ensure Open Problem Reports have Closure Plans, are Categorized Appropriately, and have no Impact on Acceptance	This task helps assess maturity of the system to be fielded. Different programs use the Discrepancy Report (DR) system differently and interpret definitions of DR categories differently. It is important to understand what the number of open DRs reveals about the maturity of the system, but it is just as important to understand how the DR system is being used and how quickly the open DRs can be worked off. For example, a DR may be assessed as a CAT 1 but the capability that is not functional may not be needed until a later phase of the transition. A program's process may also be underrating DRs and thereby hiding risk.
Ensure Oversight Includes Early Determination of Critical Metrics, Gathering of these Metrics, and Assessments Briefed to Provider and Receiver Government Organizations	The system to be fielded must be of sufficient maturity to function reliably in the operational environment. The program office needs to develop effective metrics, particularly in the area of software development, to evaluate the maturity of the products. Ensure gathered metrics reasonably depict specifics of progress towards a successful operational transition.

21.5.2 Assess Certification Readiness

There are two fundamental questions to be addressed by the assessor in this section: "Are all the required certifications complete?" and "What have we learned from other independent assessments?" This section addresses the Certification Documentation Product Group. The focus of this section includes, but is not limited to: FCA/PCA, AFMAN 63-119 certification, IA certifications, and operational test reports. This section should also address any other "independent" readiness assessments performed by other organizations (sometimes with Aerospace support). These might include contractor reviews, IRRTs, and Independent Review Teams. There is potential for conflicting findings and recommendations from this various reviews. The Aerospace Mission Assurance assessment should address conflicting information, describe open issues/showstoppers/high risk areas identified by these other efforts, and give an Aerospace recommendation/opinion on validity of those findings.

Task	Description
Ensure Independent Contractor Mission Assurance Processes are Reviewed and Assessed	If the contractor conducts independent assessments, evaluate their effectiveness. Are the reviewers credible and independent? Are the findings significant, actionable, and of value to the transition goals? Did the contractor management take the recommendations seriously? As needed, highlight areas of agreement and disagreement. This task helps assess and consolidate findings from and independent review teams for relevancy to go/no-go decisions.
Ensure AFMAN 63-119 or Equivalent Template Content has been Reviewed, Executed and Assessed	Assess the level to which the program has complied with AFMAN 63-119 or equivalent template content [1]. As needed, highlight areas of agreement and disagreement. This task helps assess findings from certification teams for relevancy to go/no-go decisions.
Ensure Program has Identified and Completed Appropriate Certifications	Ensure program is on track to procure other required certifications (e.g., Information Assurance, Failure Analysis, etc.). As needed, highlight areas of concern with technical approaches to obtain certifications. This task helps assess findings from certification and independent review teams for relevancy to go/no-go decisions. Ensure all open actions and liens from previous reviews are closed.

21.5.3 Assess Deployment Tools, Processes, and Procedures

When the development of a system is complete or nearing completion, it must be put into the operational environment for final testing and operational acceptance. The events involved with this activity are sometimes complex and increase risk to the operational system. This section addresses the "mechanics" of the transition or fielding events. The fundamental question to be addressed in this section is: "Are the special products and processes required to execute the transition sufficient and validated?" In the event of transitions with multiple phases, assessments should be done prior to each phase. The intent is to ensure that the level of operational risk induced by the transition is acceptable and that readiness activities, special transition procedures, and special transition functions/products are sufficient to successfully execute the transition. The assessment strategy should reflect the sequence of transition readiness events (rehearsals, risk reduction events, etc.) and the particular tools and products used by the program.

During the preliminary design review (PDR) and critical design review (CDR) timeframes, the emphasis should be on identifying any special transition tools, processes, or procedures that need to be developed. As the program plans its transition approach, it should be identifying operational risks or impacts that could be induced by the transition events. Early operational stakeholder knowledge and acceptance of potential impacts or outages will lay the groundwork for the later operational go/no-go decisions.

Task	Description
Ensure Adequate Planning and Analysis has been Performed on the Transition Event(s) and that, if Required, Special Operational Transition Procedures have been Developed and are Properly Vetted	Programs should be working with operational commands from the beginning to understand the sequence of decisions that must be made to proceed through the planned transition events. Often General Officer approval is required to proceed with the transition. The decisions to be made and who makes them should be defined as early as possible. Testing of unique transition components (i.e., capabilities to support the safe transition of the system into operations), should be incorporated in the program baseline. The program should have a complete transition plan which includes all steps and procedures for all elements. Details of all transition aspects affecting each legacy and new site as well as end users need to be included in the plan.
Ensure the Operational Sites can Execute Contingency and Fallback Processes, and that these Processes and Agreements are Understood and Documented by Provider and Receiver Organizations	Operations planning for the transition events should include rehearsal-like activities (for real-time operational activities as part of the transition and where the transition introduces operational risk) and contingency/fallback planning and exercises. Every attempt should be made to exercise these processes in the development or off-line systems before entry into operational test.
Ensure Provider and Receiver Organizations Agree on Risks and Impacts to Users, Assets, Operations, Acquisitions, and End Users if Transition is Rejected or Delayed	This task helps assess programs understanding of and strategy to mitigate operational risks induced by transition. Potential impacts from a failed or delayed transition could be broad and involve government, commercial, civilian, and international concerns/users.
Ensure Government, Contractor, and Operations Participants at Product Reviews take Actions, Communicate and Effect Changes and Closures, and Take Appropriate Actions to Update Documentation Regarding those Changes and Closures	Successful transitions occur when transition issues are part of the day-to-day management of the program, including level of review at major program design milestones and test events. Ensure transition planning and reviews are scheduled and executed as needed throughout the development and test cycles. Ensure engagement and participation of all affected organizations, not just operations and the development contractor. This may require additional scope for the contractors. Programs should consider scope for contractor involvement in transition planning during the request for proposal (RFP) development. This task helps assess programs understanding of and strategy to mitigate operational risks induced by transition.

21.5.4 Assess Personnel and Resources

This section addresses the readiness of the end operators to receive, operate, and sustain the system during and after the transition. The fundamental questions that are addressed are: “Is the end user ready to receive and operate the system?” and “Is a transition team in place and capable of deploying the system?”

This section applies to the operational units, and tailoring should be done to account for specific program resource needs, such as government versus contractor operators.

Early in the program, the program office needs to work with the operational command to ensure that sufficient manpower is programmed, including accommodation for test schedule delays and extensions. As the development progresses, close attention should be paid to the availability of operation personnel and the scheduling of any training that is required at each major transition milestone.

Task	Description
Ensure All Operations Personnel Complete Training and Certification, and that All Aspects of the Training Program are Sufficient to Support Continued Operations	An adequate number of trained operators must be available to support operational testing, transition events, and continued operations. This staffing should include surges for combined testing and operations, operations at alternate and backup ground locations, and extended rehearsals. Anomaly recovery and other non-nominal operations need to be included as well. Ensure training program and required skill levels have been described and are understood.
Ensure Operational Training, Logistics and other Support Functions have Requisite Skills, Products and Processes to Execution Operational Test and Operations	Ensure the training program and required skill levels have been described and are understood. Ensure organizations which support the operational sites and are directly involved in operations, such as external communications link providers, third-party maintenance and vendors, and co-hosted operational units, are staffed and trained to support both rehearsals and operations. Examples of logistics functions include security, transfer of secure and/or classified data, maintenance and sparing, and facility safety. Ensure rehearsals exercise the support organizations' functions and validate their effectiveness.
Ensure Operational Procedures to be Used by the Operator are Developed and Validated	Procedures used in developmental test and operational test many times do not have actual operational scripts, operational commands, or other test-unique entries. This includes technical orders as well as special or temporary procedures to be used during the transition. Depending on the robustness of the test environment, the test teams may not have any opportunity to execute the actual operational procedures—procedures which require no modifications for use after transition or launch. Off-line and analysis software can be susceptible to failure, having only used test data before launch. Special emphasis should be placed on the procedures used for those tools, including ephemeris calculations, calibrations, and analyses. Fall-back procedures are often written, but seldom exercised. Review and use of these procedures during developmental test is essential to reducing risk during operational test.
Ensure Plans are in Place to Sustain (Including an Emergency Fix Process) the Operational Products from Initial Fielding to Eventual Transition to the	Ensure the system sustainment plans and procedures are tested and executed by the operations and sustainment staff during developmental test and operational test. The prime example is ensuring who the software maintainers who will fix discrepancies after operational acceptance are properly trained and familiar with

Task	Description
Final Sustainment Organization	the software. This task helps assess readiness to sustain the system after operational acceptance.
Ensure Technical Support Personnel (e.g., Factory Tech Advisors) have been Properly Trained/Familiarized	This task helps assess readiness of support personnel to sustain and operate the system after operational acceptance. Support personnel in this case are the on-site system (space and ground) experts that are called upon to resolve anomalies and plan unusual and complex operations that are beyond the capability of the crew force. Sustainment plans and budgets must be in place to ease the later transition of the system into sustainment. Ensure these support personnel are a part of the operational test team, and that they understand operational unit policies and procedures related to the new system prior to the transition.

21.5.5 Assess Transition Planning

This section addresses how well the program has planned, prepared and executed its transition. This area assesses documentation, organization, command and control, resources and other transition-related products. It will also assess the application of the transition principles across the program. These are cross-cutting principles, dealing with how programs integrate their transition planning into the overall program. A fundamental question to be asked is: "To what extent has the program incorporated transition planning and preparation into its processes?"

High-risk transitions, or transitions with significant programmatic implications, should have the most robust and detailed assessments. Each program will have a unique set of elements to be tested, as well as integrated segment or system tests. These tasks should be tailored to cover any of the individual element, segment, and system test programs that bear directly in lowering transition risk during developmental test. Assessments should focus on the transition risk mitigated in a particular test, rather than requirements sell-off. In a multi-phased deployment, each phase of the deployment should have its own set of tailored tasks.

Task	Description
Ensure all Stakeholders Participate in Planning, Reviews, and Acceptance Milestones/Events, Starting Early in the Life-Cycle, and they understand the Impacts of the Acceptance	This task assesses how the program plans and manages the transition. The plan should be developed early, and be reviewed continuously through the program, especially when major program re-planning occurs. A government transition director should be appointed to oversee transition readiness and integrate all the transition planning between stakeholder organizations. Transition planning should be considered during the requirements development phase to ensure the transition is planned into the program and, if necessary, requirements established. The design of the system must be considered for its ability to facilitate a smooth transition. Stakeholders need to participate in the transition planning, preparation, and execution activities to ensure common understanding and to support expectation management.

Task	Description
Ensure the Program has Established Regular Forums for Planning Transition Activities and Resolving Transition-Related Issues	This task assesses the process for communicating transition information and provides the means for monitoring and modifying transition activities as needed. Programs should establish regular forums to address transition issues and ensure that aspects of the transition are integrated into program planning. These forums provide a structured, recurring means to discuss activities, resolve issues, and enable critical community-wide communications. The meetings should begin at program initiation and continue until system transition. The frequency of the meetings at the beginning of the program is dependent on the transition requirements of the program but quarterly should be sufficient. As a program moves into testing, these meetings should be held monthly. All stakeholders affected by transition, including end users, should be represented in these forums.
Ensure all Issues Regarding Operational Transition are Surfaced to Provider and Receiver Organization	This task assesses the ability of a program to identify, prioritize, and resolve issues as early as possible to reduce program costs and risk. Open communication between provider and receiver organizations is critical to program success. Early identification of issues and the subsequent resolution increases program effectiveness, confidence, trust, and supports expectation management. Having regular forums and an established process to resolve issues or bring them to the surface supports accomplishing this task.
Ensure Appropriate Expectations are set Relative to the Scope and Reliability of the System	This task assesses the existence and maintenance of mutually agreed upon expectation management agreements (if the program has set them up). Expectation management is critical to program success. Expectation management requires communication and understanding between the acquisition and user community. Often requirements for space systems evolve during the development of the system. To support expectation management, a best practice is to formally document agreements between the acquisition and user organizations.
Ensure Appropriate Assessment Points are Defined, and that Relevant Artifacts are Created at those Points	Assessments should be performed at milestones defined by the program or at annual intervals to ensure issues can be resolved as early as possible to reduce costs and program risk. The Aerospace program office should identify points in the program to provide assessment briefing to senior management on operational readiness.
Ensure Entry/Exit Criteria for each Program Milestone are Agreed To by All Stakeholders and are Achievable on the Planned Schedule	Establishing and meeting entry/exit criteria for program milestones is critical to ensuring programs are on track. Ensure provider and receiver management understand impacts of missed milestones, and risks identified. Deviations from entry/exit criteria must be documented as liens and adjudicated appropriately.

21.5.6 Aerospace President's Review (APRs) for System Transitions

System and ground transitions often have attention at the highest levels of military and The Aerospace Corporation. An APR or Vice-Presidents Review is

often directed to equip Aerospace senior management to advise their government counterparts on the readiness status and risk of upcoming system transitions. For those programs that Aerospace does not have direct accountability, the principle content of these mission assurance tasks should be applied at senior management review as one of the last independent review gates prior to going operational.

Programs that have held APRs for system transitions have generally included the following topics in their reviews. Each of the topics relates back to one or more of the assessment areas just described.

Task	Description
Overview of the Transition Event	A transition event refers to the set of activities in the operational environment required to bring the new system on-line. Providing senior leadership adequate background and overview material is critical. If possible, the program should schedule an information session early in the process to prepare them for later decision briefings. Where APRs for launches are very similar in technical content, the content of ground or system transition events can vary widely. These transitions are highly interconnected to the specific missions of the program as well as the architectures of the systems involved. Senior leaders may not be as familiar with these missions and architectures as they would be with launch vehicles and spacecraft hardware.
Government Processes	AFSPCI 10-205 provides high-level guidance for operational transition processes. However, the implementation will be unique for each program. The ultimate success of a transition will depend greatly on how the government has structured transition planning and how effective the planning is. The APR should present material describing the role of the program's transition director, the involvement of all the stakeholders including higher headquarters, the level of engagement of senior government leaders, and the effectiveness of the overall government process.
Aerospace Accountability and Mission Assurance	An APR for a system transition should provide a description of the Aerospace accountability and level of STE across the life of the program. The roles of each of the groups in the program office (ground, system engineering and integration team (SEIT), operations) as well as significant Engineering and Technology Group (ETG) contributions should be highlighted. A status of the integrated mission assurance tool (iMAT) assessment tasks should be provided. At an APR or preliminary meeting, not all of the tasks may be closed. An overview of the approach, taking into account allocation of Aerospace resources, analyses and assessments performed, and risk assessments using the mission risk assessment Technical Procedure document should be provided. Ideally, this will be an overview of the mission assurance work plan and results as documented in iMAT. Risk assessment discussions should include evidence that identified risks were correctly mitigated and identification of any unanticipated critical risks that were raised.

Task	Description
	Individual risks can be briefed as a group as part of the mission assurance presentation or briefed individually in the appropriate technical area and summarized at the end of the presentation.
System Testing and Requirements Verification	A good description of system testing leading up to the transition event should be provided. This should include, but not be limited to, an account of the requirements verification process. More importantly, the APR needs to convey the kind of testing that was done, how operationally realistic it was, and how it mitigates the risk of operating the system in the operational environment.
Readiness to Enter Operational Test	The APR should include a section describing the program's readiness efforts to enter operational test. This would include how the program implemented the AFMAN 63-119 readiness templates. In some programs the Aerospace team has minimal involvement in the assessment of the templates, however the Aerospace team must be aware to the 63-119 template contents and brief an Aerospace assessment of readiness to enter operational test.
Product Readiness	The APR briefing should develop significant attention to the readiness of the systems to be fielded. In most cases, this will entail a detailed description of the software development and test process used and the Aerospace assessment of the maturity of the product. Metrics like defect rates and open discrepancies should be discussed.
Readiness of Impacted Systems	These types of transitions can potentially affect other parts of the system or mission. End users may have had to make changes to their system to interface with the system being fielded. Spacecraft may need to be changed or operated in a different way during the transition. The APR presentation needs to show that all impacted parts of the system are ready for the transition.
Operations Readiness	A section of the APR should be dedicated to describing how the operations personnel have prepared for the transition. This can include training and rehearsal activities. The role of Aerospace personnel on site during the transition events should be described.
Summary of Open Issues and Recommendations	Finally, a summary of open issues and the Aerospace recommendation to proceed or delay until corrective actions are taken is presented.

21.6 Key Lessons Learned

This section describes several negative examples of poor transition practices. Program names are not used to avoid possible operations security concerns and to focus attention on the transition practice, not the individual programs.

21.6.1 Communications Link Not Tested during DT

One program failed operation test because the development test effort did use an operationally representative communications link. Remote sites fed data into a central processing center. The links provided proved to be inadequate for the

bandwidth required. The developmental test focused on the processing at the processing center and did not do an end-to-end check out of the entire system.

21.6.2 System Does Not Meet User Expectations

A satellite and ground system went successfully through OT but was not accepted operationally because it was not what the users wanted. This resulted from a failure to set expectations at the beginning of the program and an ongoing lack of engagement with the user community as the system was developed.

21.6.3 Post Acceptance Development of Transition Capability

A large ground system was accepted by the government but it could not be transitioned into operations because it lacked the capability to keep the satellites in mission during the transition. The transition was delayed while additional development was done to lessen the effect of the transition. The requirements for additional system capability required to execute a smooth operational transition must be identified at the beginning of the program and transition functionality must be treated as any other system functionality throughout system development and testing.

21.6.4 Insufficient Linkage between DT and OT Programs

A radar system failed OT because the DT was not reflective of how the system was going to be tested operationally. Agreement needs to be reached early between the developers and the operational testers about how to interpret and test the requirements.

21.6.5 High Number of Software Discrepancies

A large ground system experienced numerous delays because one of its subsystems exhibited a very high number of discrepancy reports (DRs). The problem was exasperated by an improper use of the DR priority system. Many of the DRs were binned in the highest priority category even though they had acceptable workarounds. This experience points to the need for good and accurate development metrics so the maturity of a system can be evaluated at the time of fielding.

21.7 Summary

Successfully fielding a new system, particularly when the fielding impacts an operational legacy system, can be a challenging and risky undertaking. The most critical factor for success is early and continued focus on the fielding activities

and requirements over the life of the program. This type of focus will lead to test planning and execution that not only sells off requirements, but ensures the system will perform successfully in the operational environment. Early focus may also identify design features that can save cost and schedule during the later phases of the program. Finally, early and persistent effort ensures stakeholder buy-in, avoiding last minute crises from unexpected "no-go" votes at key milestones.

21.8 References

1. Air Force Manual (AFMAN) 63-1119. *Certification of System Readiness*. June 2008.

2. McCasland, David H., Jeffrey J. Vance, and Wayne G. Yenne. Assessing and Preparing for Air Force Systems' Operational Readiness. TOR 2013-00219. The Aerospace Corporation, El Segundo, CA. 2013.

3. Perry, Jessica S., Graham S. Arnold, David W. Bart, Michael W. Fortanberry, Gail A. Johnson-Roth, Norman Y. Lao, David H. McCasland, Rebecca McKenna, Kenneth R. Sieck, Jacqueline M. Wyriwitzke. *Mission Assurance Baseline, Version 2.7*. ATR-2015-00618. The Aerospace Corporation, El Segundo, CA. October 30, 2014.

21.9 Bibliography

Air Force Space Command Instruction (AFSPCI) 10-205. *Operational Transition Process*. December 2013.

Ibrahim Awwad and Bruce Arnheim "Implementation of Mission Assurance Processes for Air Force Space Systems' Operational Transitions" Space Ops 2014. 13th International Conference on Space Operations, AIAA, May5-9, 2014, Pasadena, CA.

21.10 Acronyms

AFMAN	Air Force manual
AFOTEC	Air Force Operational Test and Evaluations Center
AFSPC	Air Force Space Command
APR	Aerospace President's Review
CDR	critical design review
DR	discrepancy report
DT	developmental test
ETG	Engineering and Technology Group
FCA	functional configuration audit
HQ	headquarters

IA	information assurance
iMAT	integrated mission assurance tool
IRRT	independent readiness review team
OT	operational test
PCA	physical configuration audit
PDR	preliminary design review
PEO	program executive officer
PO	program office
RFP	request for proposal
SEIT	system engineering and integration team
SMC	Space and Missile Systems Center
STE	staff years of technical effort
TLYF	test like you fly

Chapter 22
Mission Operations

K. Rex Childers
Operations and Sustainment
MILSATCOM Division
James W. Boswell
Systems Engineering
Imagery Programs Division

22.1 Introduction/Background

Mission operations encompasses command and control of the space and ground assets with respect to day-to-day, non-routine, and anomaly resolution operations. Mission operations is closely linked to mission management (MM) as it carries out the activities planned in MM. Mission operations consists of people, processes, plans, software, hardware, and procedures that may also be shared with mission management. Mission operations is responsible for conducting space/ground asset command and control, as depicted in the figure showing the ground segment reference architecture. Mission operations begins during pre-launch with the development of the operations concepts (OPSCON), procedures, command plans, and crew training necessary to operate the space and ground assets safely and efficiently. Mission operations continues throughout the life of the system, including retirement and disposal. Over the life of the system new products may be developed and/or existing products may be refined as lessons are learned and applied. The responsibility for operations may transition from one organization to another.

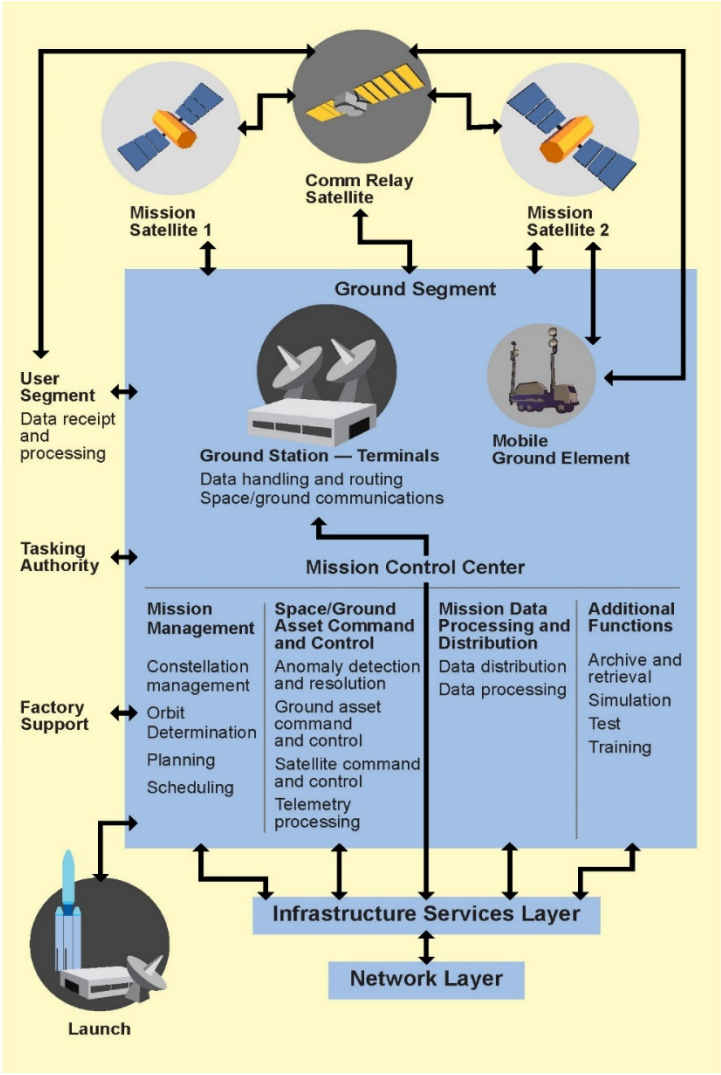


Figure 22-1. Ground segment reference architecture.

22.2 Definitions

Anomaly System event which either threatens system safety or causes degraded performance. An anomaly may also refer to nonconformance to expected performance that may require reconfiguration to resolve.

Anomaly resolution operations The diagnosis and recovery efforts by the operations team. May require detailed review of current and past telemetry, development, and execution of new and/or unique command procedures.

Command procedures Step-by-step sequences of directions, commands, expected satellite telemetry indications, and logic to accomplish a specific activity on the space vehicle. These maybe online or on separate media, such as paper.

Ground procedures Step-by-step instructions, expected ground system indications, and logic to accomplish a specific activity on the ground system. These maybe online or on separate media, such as paper.

Operations concepts (OPSCON) Documented guidelines of how various aspects of operations should be performed, as well as the roles and responsibilities of the various organizations and personnel involved.

Operators Users of the system responsible for configuring the ground system and command and control of the satellite.

Operations engineers Users of the system responsible for providing satellite system and/or subsystem and ground system expertise in support of operations.

Operations team The combined group of operators and operations engineers that support operations.

Hazardous/restricted commands Commands that can cause serious impacts on the satellite if sent inadvertently. (These commands generally require two steps.)

Telemetry/status displays The windows or full screens that present telemetry/status to the users.

22.3 Description of Mission Operations

It is best to begin planning for mission operations early on in the acquisition lifecycle. In fact, it is highly recommended that mission operation activities take place concurrently with the acquisition and development lifecycle. Figure 22-2 shows an approach to aligning the mission operations activities with the acquisition lifecycle.

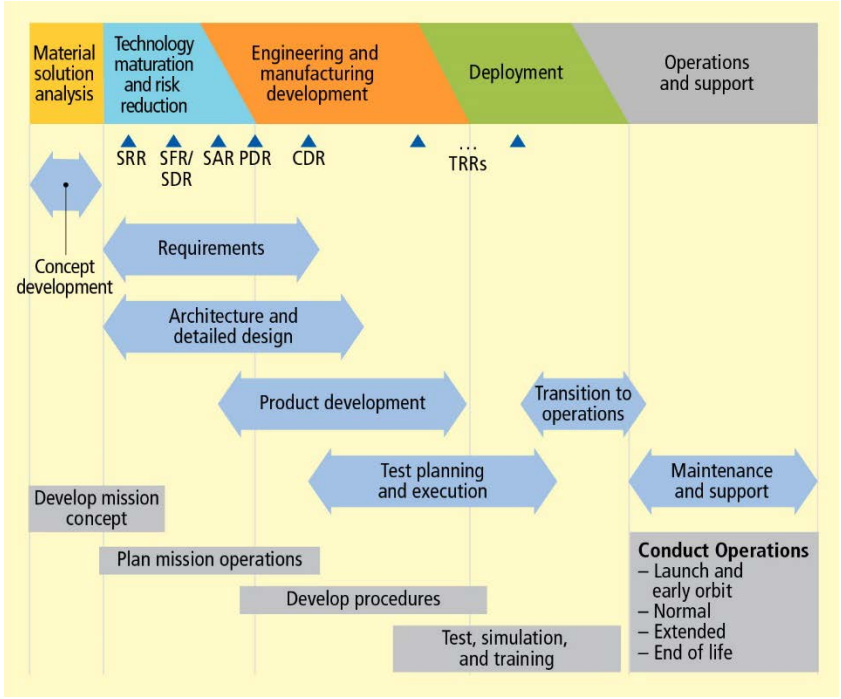


Figure 22-2. Alignment of mission operations lifecycle with acquisition lifecycle.

22.3.1 Develop the Mission Concept

The first and most important step in mission operations is to define the OPCON. The main purpose of the OPCON is to document the strategy for how the mission (Msn), space vehicle (SV), payload (P/L), and the ground segment (GS) will be operated throughout the entire mission lifecycle. This document should also describe the mission architecture to include a description of the SV, P/L, and GS along with a definition of all mission phases. The mission operations development team is responsible for developing this concept. The mission operations development team consists of personnel from operations engineering, as well as personnel from S/C, P/L, and GS operations. To develop a mission operations concept, the following steps may be followed: [1]

1. Determine mission objectives, requirements, constraints, and type of mission
2. Characterize mission concepts and associated space mission architectures:
   (a) Information and telemetry system characteristics
   (b) Payload characteristics

(c) Space vehicle characteristics
(d) Data products
(e) Ground system characteristics
3. Allocate resources to functions for each mission phase
4. Document the results of the analysis

The operators are not always invited to participate in the mission operations concept selection, as this is the basis for the acquisition of the space vehicle or payload. Depending on the acquisition authority, the GS operations may influence the baseline mission concept, participate in the development of alternative mission operations concepts; perform trade studies; and contribute to the assessments of the mission utility, lifecycle cost, relative complexity, and cost of operations. In these cases, the results of these analyses should be documented with justification as to the choices selected.

22.3.2 Plan Mission Operations

The next mission operations activity is to generate a top-level description or plan which spans the mission's lifecycle. This plan describes how the mission will be flown, expresses objectives in operational terms, and sets in place all major activities. The mission plan is consistent with and generated after the mission operations concept is developed and should contain the following:

1. Mission objectives and goals
2. Orbit or trajectory description and orbital plan
3. Payload description and operation
4. Space vehicle bus description and operation
5. Mission phases
6. Description and techniques of mission operations
7. Mission rules and method of verification

22.3.3 Develop Procedures

Procedure development is perhaps the most critical piece of the mission operations lifecycle. Procedures are developed for space vehicle contacts, to ensure data transfers take place for all uplinks and downlinks, to monitor all systems during the contact, for long-term and short-term planning in support of mission operations, for anomaly planning and commanding, for launch and early on-orbit deployment, for checkout, and for initialization, and to document all activities that occur during the contact [1]. The mission planning and the command and control of subsystems provide functions to allow the operations engineers to develop automated, as well as manual, space vehicle procedures. The procedures tell mission operators what to do before, during, and after a contact with the space vehicle.

22.3.4 Test, Simulation, and Training

Once the procedures are developed, space vehicle simulators are used to test the logic and the accuracy of the procedures. This testing will take place while the operations crew is being trained to operate the ground segment. Once this activity is complete, the operations crew is ready to begin launch and early orbit checkout.

22.3.5 Conduct Operations

At the end of the launch and early orbit checkout phase, transition begins to nominal operations or initial operational capability (IOC). At the end of IOC, normal operations begin. The main operations activities are: verify and maintain space vehicle and payload health, reconfigure and command the space vehicle and payload, and detect, identify, and resolve anomalies.

The steps to accomplish these activities are: [1]

1. Configure the ground segment to support passes. (A pass is when the space vehicle is visible and in range for contact.)
2. Transmit commands to the space vehicle
3. Verify the receipt of commands
4. Monitor the space vehicle's health and safety
   (a) Compare predicted and actual space vehicle states
5. Coordinate mission control functions
   (a) Planning and scheduling before and after passes
6. Generate an integrated plan for as-flown activities
   (a) Post pass report
   (b) Document deviations from pass plan
7. Support activity planning and development
8. Generate operations schedules and plans for future passes
9. Negotiate and schedule tracking support
   (a) Generate pass plans
10. Support planning and analysis teams
    (a) Investigate anomalies

22.4 Technical Considerations

From a mission-operations perspective the ground segment design considerations should focus primarily on the needs of the operations team to conduct operations within physical, technological, and funding resource constraints. The following operational areas need to be considered when designing the ground segment:

- Operator workstations:
  - The number of workstations available for the users depends on the scope of the mission. Anomaly resolution and complex activities may require a significant number of monitors to support parallel viewing. Routine mission operations may not. Physical floor space constraints may apply.
  - Workstation makeup, including both hardware and software, may vary depending on the primary purpose of the workstation. For instance, orbit analysis or telemetry trending analysis may not require the same number of monitors or the same software capability as a real-time command and control workstation. The number of monitors required per workstation may be physically and/or capacity constrained as well.
  - Human systems integration needs to be thought out so each workstation is as intuitive to the user as possible to ensure safe and consistent operations.
- Operator input safeguards:
  - Should be built-in to prevent inadvertent errors in configuration or commanding.
  - Display of the command procedure and/or commands should be visible to other participants in the activity to allow independent verification of the command prior to transmission.
  - More than a single action should be required for the operator to transmit hazardous/restricted commands.
  - Display of more than just the command name to the user(s) should be visible prior to transmission to verify correct name and description (for commands built during planning that have unique names and descriptions).
- Online command or configuration procedure execution:
  - If the ground system is designed for online procedure execution the ability to execute automatically or semi-automatically should be available.
    - The use of automatic or semi-automatic execution is determined primarily by the operations concept.
    - The operator at execution time should select the execution mode to allow a procedure designed for automatic operation to be executed semi-automatically to control the pace in non-routine circumstances
  - The software that controls the execution of a procedure should also allow for the built-in pauses in the procedure to allow for operator review of data and/or input that could not be pre-determined in mission planning.
- Satellite telemetry screens:
  - Should be logically organized by subsystem/activity.
  - Should be relatively easy to navigate.

- Should be arranged hierarchically.
- Should be complete in the sense that operators/operations engineers have access to the information necessary to determine satellite status and diagnose problems independent of command procedures.
- Telemetry-limit display schema should alert users to out-of-limit conditions normally displayed as yellow and red. These should be highlighted on the telemetry screen where the out-of-limit point is displayed with some indication on higher-level screens to point the way to the lower-level screens as appropriate. Alarm Indications may also be presented on a separate alarm/warning/error display.
- Concerns with the use of graphical-widget-based versus text-based displays:
  - Display space: Graphical widgets tend to take up more display space per individual telemetry point than text.
  - Telemetry/status point population: The larger the population the more display space required.
  - Scope of operations: Anomaly diagnosis may require access to more telemetry points/displays than routine operations.

- Ground system configuration/status displays:
  - Should be logically organized.
  - Should be relatively easy to navigate.
  - Should be possibly arranged in hierarchically.
  - Should be complete in the sense that users have the information necessary to configure ground assets, determine status, and troubleshoot problems.
  - The use of graphical widgets here is desirable.
- Command and telemetry history archiving:
  - The ground system should have the ability to store and playback telemetry for purposes of trending and near-real-time analysis of data.
  - Command history should be stored and available for analysis to assist in the determination of possible commanding issues.

22.5 Programmatic Considerations

Listed below are some lessons learned for ground segment acquisition offices to consider concerning mission operations:

- Understand fully the scope of the operations to be conducted using the ground system. Systems intended to be used primarily for complex operations and anomaly resolution will need to be much more robust than those intended primarily for routine operations in terms of commanding that can be accomplished, telemetry that can be displayed, and flexibility in executing command procedures and/or commands.

  - Example: Early in development of the system the development team was focused on the needs of the operators in accomplishing routine operations. However, the legacy system which was to be replaced by a new ground system was used primarily for launch and early orbit operations, other complex operations, and anomaly resolution. Once this was communicated to the development team the scope of the design was refocused, which led to changes in the number and type of workstations needed and an increase from 25% to 100% of the telemetry points to be displayed on pre-designed screens to make the system more robust.
- Involve the targeted users of the system, both operators and operations engineers, in the development of the mission operations element to ensure operations needs are fully understood and, if possible, implemented. These users should be experienced in the intended or similar missions, and with the role of ground systems in operations. The availability of knowledgeable operators and operations engineers is key to support the legacy ground system replacement.
  - Example 1: Due to an ongoing launch and early orbit campaign operations personnel available to support the new ground system development were very limited in terms of number and experience. As a result, when the users began actually using the system they found it fell short of meeting their needs. A significant amount of rework and schedule delay was required to finally meet user needs.
  - Example 2: Experienced military and contractor operators were allocated to support the development of the ground system. They participated and provided expert input to virtually every aspect of the design and implementation of the operations requirements. As a result, the system met requirements without significant rework or schedule delay. Inputs by experienced operators and operations engineers supporting the development resulted in the system meeting user needs upon delivery.
- If the new ground system is to replace an existing ground system then the developers and users should temper the desire to take full advantage of new capabilities with the need to minimize the change to operations concepts and procedures that are proven on the legacy system. This is especially true if the intended users cannot provide adequate or knowledgeable support to the development effort. It should not be assumed that the developer can develop the mission operations element without this kind of assistance. It may be better to ease into the new capabilities as the users become more familiar with the system and can articulate their needs more accurately.
  - A good example of this was when development efforts to replace the legacy ground system in support of two different space vehicle systems took markedly different approaches in terms of command procedures and telemetry displays. One space system development

preceded the second space system development in terms of the acquisition timeline.

The first system required all new telemetry screens using a mix of graphical widgets and text. Legacy paper command procedures were essentially copied line-for-line into the online procedure application. Even though the second space system had a knowledgeable team to represent the users, it was not large enough to support the detailed redesign of all the legacy command procedures or telemetry displays. Therefore the text-based telemetry displays of the legacy system were recreated with only minor modifications. Only a few of the legacy command procedures determined by the operations team to be most amenable and useful to put online were converted into online procedures. The remainder of legacy command procedures remained paper-based and were only modified slightly to account for the new ground system unique attributes. An online procedure to allow the operator to enter commands manually per the paper command procedures was employed to replicate the legacy functionality for commanding.

As a result, the first system's online procedures having not been designed to run online were declared unusable by the operators and largely scrapped. However, in order to use the legacy paper procedures efficiently all of the telemetry displays from the legacy system had to be converted similar to what was done on the second space system. All of this delayed the transition from the legacy system to the new ground system. For the second space system, copying the telemetry screens from the legacy system resulted in the legacy paper command procedures being usable with only slight modifications. This also saved a significant amount of development and testing time for telemetry screens. In part, this approach allowed the government to meet the schedule for decommissioning of the legacy system.

The lesson learned here is to start planning mission operations early in the development cycle. Operations needs to be closely considered in the conceptual and detailed design of the space vehicle as well as the ground systems [2]. Having an easy-to-use ground segment and a space vehicle that recovers well from failures will decrease operator workload and decrease operations and maintenance costs.

22.6 References

1. Wertz, James R., David F. Everett, and Jeffrey J. Puschell, (editors) *Space Mission Engineering: The New SMAD (Space Technology Library*, Vol. 28). Microcosm Press, July 2011.

2. Squibb, Gael, Daryl Boden, Wiley Larson, Cost-Effective Space Mission Operations. McGraw Hill, 2006.

Chapter 23
Maintenance and Sustainment

C. Jean J. Wang
Software Acquisition and Modeling Department
Software Engineering Subdivision

23.1 Introduction

A major part of a ground segment's lifecycle is the sustainment phase and the integration of sustainment activities during this period. The integration process includes the hardware and software of a space system ground segment.

23.1.1 Acquisition Process

There are three acquisition stages which are broken down to five acquisition phases:

1. Pre-systems acquisition, including:
   (a) Materiel solution analysis (MSA) phase
   (b) Technology maturation and risk reduction (TMRR) phase
2. Systems acquisition, including:
   (a) Engineering and manufacturing development (EMD) phase
   (b) Production and deployment (P&D) phase
3. Sustainment, including:
   (a) Operations and support (O&S)phase

An acquisition phase consists of all the tasks and activities needed to bring a program to the next major milestone occur during an acquisition phase. Phases provide a logical means of progressively translating broadly stated capabilities into well-defined, system-specific requirements and ultimately into operationally effective, suitable, and survivable systems.

O&S is the lifecycle phase for maintaining and sustaining delivered systems certified at the P&D phase. O&S is the fifth and final lifecycle phase of the lifecycle [1]. This phase consists of two efforts, lifecycle sustainment and disposal. The phase is not initiated by a formal milestone, but instead begins with the deployment of the first system to the field, an act that initiates the lifecycle sustainment (LCS) effort of this phase. The LCS effort overlaps the full-rate production and deployment effort of the P&D phase. Refer to Department of Defense Instruction (DODI) number 5000.02, for a full and detailed description of the Operation of the Defense Acquisition System [1]. Figure 23-1 is extracted from DODI 5000.02; it illustrates the generic interaction between the capability requirements process and the acquisition process.

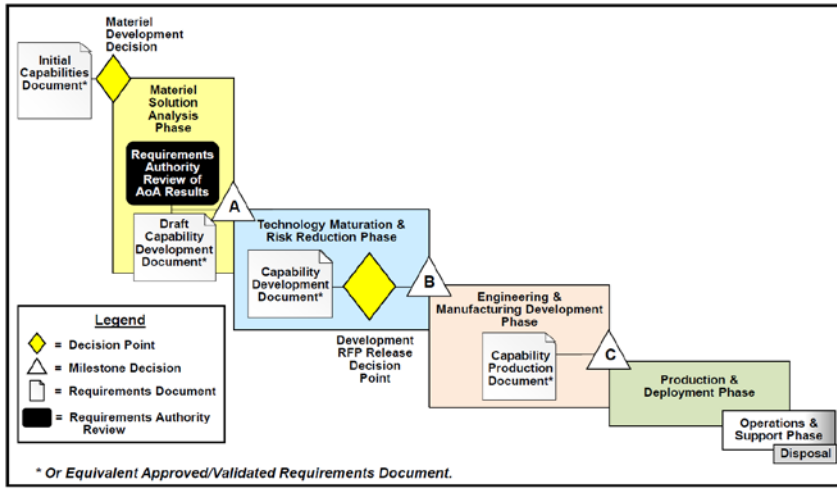


Figure 23-1. Capability requirements and the acquisition process [1].

For a defense acquisition program, lifecycle cost (LCC) consists of research and development (R&D) costs, investment costs, operating and support costs, and disposal costs over the entire lifecycle. These costs include not only the direct costs of the acquisition program, but also indirect costs that would be logically attributed to the program. In this way, all costs that are logically attributed to the program are included, regardless of funding source or management control.

The costs incurred during the O&S phase of a weapon systems' lifecycles have been rising while defense budgets have been falling. This situation has caused the Department of Defense (DOD) to continue to improve product support with a specific focus on increasing readiness and enabling better cost control. It also resulted in many updated directives, instructions, guidebooks, and handbooks all focused on improving the lifecycle sustainment process. The purpose of these efforts is to establish a more efficient, cost-effective means of support during a system's post-interim operational capability (IOC) sustainment period.

The Defense Acquisition Guidebook (DAG) "Life-cycle Logistics," provides the guidance for the program manager (PM), product support manager (PSM), and lifecycle logisticians [2]. The key is to involve the PSM and logisticians in the earlier acquisition lifecycle phases ensure that the system requirements, architecture, and design meet the supportability needs, i.e. effective and timely product support capability to achieve the system's materiel readiness and to sustain operational capability.

23.2 Definitions

There are terms and their definitions that are useful for understanding sustainment and maintenance sourced from the *Glossary: Defense Acquisition Acronyms and Terms* [3].

Acquisition category (ACAT) Categories established to facilitate decentralized decision-making and execution and compliance with statutorily imposed requirements. The categories determine the level of review, decision authority, and applicable procedures. See DODI 5000.02 for a definition of the program ACATs and types and their acquisition requirements [1].

ACAT ID The Milestone Decision Authority (MDA) is the Under Secretary of Defense for Acquisition, Technology, and Logistics USD (AT&L) [4]. The "D" refers to the Defense Acquisition Board (DAB) at major decision points.

Baseline Defined quantity or quality used as a starting point for subsequent efforts and progress measurement: can be a technical, cost, or schedule baseline.

Configuration A collection of an item's descriptive and governing characteristics, which can be expressed in functional terms, i.e., what performance the item is expected to achieve; and in physical terms, i.e., what the item should look like and consist of when it is built.

Deploy/deployment Fielding a weapon system by placing it into operational use with units in the field/fleet.

Key performance parameters (KPP) Performance attributes of a system considered critical to the development of an effective military capability. A KPP normally has a threshold representing the minimum acceptable value achievable at low-to-moderate risk, and an objective, representing the desired operational goal but at higher risk in cost, schedule, and performance. KPPs are contained in the capability development document (CDD) and the capability production document (CPD) and are included verbatim in the acquisition program baseline (APB). KPPs are considered Measures of Performance (MOPs) by the operational test community.

Key system attributes (KSA) Attributes or characteristics considered essential to achieving a balanced solution and/or approach to a system, but not critical enough to be designated a KPP. KSAs must be measurable, testable, and quantifiable. Mandatory KSAs are specified by the joint capability integration and development system (JCIDS) manual [5,6]; other KSAs may be specified by the JCIDS document sponsor lifecycle cost.

Maintenance The action necessary to retain or restore an item to a specified condition.

Metrics Parameters or measures of quantitative assessment used for measurement, comparison, or to track performance or production.

Product baseline Documentation describing all of the necessary functional and physical characteristics of the configuration item (CI); the selected functional and physical characteristics designated for production acceptance testing; and tests necessary for deployment/installation, operation, support, training, and disposal of the CI. The initial product baseline is usually established and put under configuration control at each CI's critical design review (CDR), culminating in an initial product baseline at the system-level CDR. The system product baseline is finalized and validated at the physical configuration audit (PCA).

Product support package (PSP) The integrated product support (IPS) elements and any sustainment process contracts or agreements used to attain and sustain the maintenance and support concepts needed for materiel readiness.

Product support strategy The business and technical approach to design, acquire, and field the product support package to execute the sustainment strategy. It begins as a broad concept and evolves into a detailed implementation plan documented in the lifecycle sustainment plan (LCSP).

Supportability A key component of availability. It includes design, technical support data, and maintenance procedures to facilitate detection, isolation, and timely repair and/or replacement of system anomalies. This includes factors such as diagnostics, prognostics, real-time maintenance data collection, and human systems integration (HIS) considerations.

23.3 Key Tasks/Principles

A key lifecycle management enabler, IPS, is the package of support functions required to deploy and maintain the readiness and operational capability of major weapon systems, subsystems, and components, including all functions related to weapon systems readiness. The package of product support functions related to weapon system readiness, which can be performed by both public and private entities, includes the tasks that are associated with the IPS elements which scope product support.

23.3.1 Integrated Lifecycle Management

Air Force Instruction (AFI) 63-101/20-101, Acquisition/Logistics Integrated Life Cycle Management (ILCM) [7] contains the overarching processes and

procedures required for execution of a program. This instruction describes the implementation of the integrated lifecycle framework, the ILCM chain of authority, roles and responsibilities of those involved, the processes, and systems engineering.

The PSM, either full-time or part-time, is an essential role to perform maintenance and sustainment related activities. The PSM is responsible for managing the package of support functions required to field and maintain the readiness and operational capability of major weapon systems, subsystems, and components, including all functions related to weapon system readiness, in support of the Program Manager's (PM's) lifecycle management responsibilities. The PSM reports directly to the PM, takes direction from the PM, and collaborates with other functional leadership within the program. The PSM is the point of contact for overall product support throughout the entire acquisition lifecycle and is accountable for all product support matters, e.g. cost, schedule, performance, and supportability.

It is important that the system is designed to be supportable to minimize the demand for product support after delivery. After system delivery, the product support must be effective and efficient. Systems must be designed so that the required product support resources will be minimal while ensuring the system still meets user requirements and needs. AFI 63-101/20-101 defines the product support requirements, e.g. LCSP, contractor logistic support, technical orders (TO), or sustainment metrics [7].

Air Force Pamphlet 63-128 provides guidance and recommended procedures for implementing ILCM and contains details for product support and sustainment [8,9].

23.3.2 Lifecycle Sustainment Plan (LCSP)

Lifecycle sustainment translates force provider capability and performance requirements into tailored product support to achieve specified and evolving lifecycle product support availability, reliability, and affordability parameters. Lifecycle sustainment considerations include supply; maintenance; transportation; sustaining engineering; data management; configuration management; HSI; environment, safety (including explosives), and occupational health; and protection of critical program information and anti-tamper provisions, supportability, and interoperability. Initially begun during the MSA phase and matured during the TMRR phase, lifecycle sustainment planning spans a system's entire lifecycle from the MSA phase to disposal.

The lifecycle sustainment plan addresses a program's product support strategy for accomplishing the supportability objectives across the lifecycle, including

during the O&S phase. The LCSP is the key logistics acquisition deliverable required in the defense acquisition system.

An LCSD is a program management tool to align and to help integrate the product support stakeholders' efforts for formulating, implementing, and executing the program sustainment strategy. The focus of these efforts is to satisfy the warfigher's sustainment requirements through the delivery of an affordable lifecycle product support package. The PSM is responsible for ensuring the LCSP documents the plan for formulating and executing the product support strategy. The design and every facet of the product support package (including any support contracts) must be integrated and will contribute to the warfighter's mission requirements. The LCSP documents the current program plan relative to sustainment and articulates the product support strategy. It is a living document describing the sustainment approach and resources necessary across the lifecycle. It must be kept relevant as the program evolves through the acquisition milestones into sustainment. The LCSP is updated to reflect the evolving maturity of the product support strategy at each milestone, at full-rate production (FRP), and prior to each change in the product support strategy, or every five years, whichever occurs first.

The LCSP outline emphasizes early-phase sustainment requirements development and planning, focuses on cross-functional integration (most critically with systems engineering), and highlights key sustainment contract development and management activities. A LCSP sample outline is available to help programs generate their specific LCSPs [10].

23.3.3 Integrated Product Support (IPS) Elements

The key lifecycle management enabler, IPS is the package of support functions required to deploy and maintain the readiness and operational capability of major weapon systems, subsystems, and components, including all functions related to weapon systems readiness. The package of product support functions related to weapon system readiness, which can be performed by both public and private entities, includes the tasks that are associated with the IPS elements which scope product support.

The following are the twelve (12) IPS elements. The details such as standard definitions, breakdowns for each of the IPS elements and sub-elements, key activities and products for each IPS element, are detailed in the *IPS Element Guidebook* [10,11].

1. Product support management
2. Design interface
3. Sustaining engineering
4. Supply support

5. Maintenance planning and management
6. Packaging, handling, storage, and transportation
7. Technical data
8. Support equipment
9. Training and training support
10. Manpower and personnel
11. Facilities and infrastructure
12. Computer resources

To emphasize the importance of sustainment, product support management, and sustaining engineering, the IPS Element Guidebook expands upon the traditional ten integrated logistic support (ILS) elements [12]. The 12 IPS elements are integrated throughout the lifecycle by supportability analysis, which begins with design interface and carries on through to sustaining engineering. The product support management element provides the framework for the integration of the other eleven IPS elements to ensure the product support solution delivered to the warfighter is fully integrated and meets the warfighter's needs in terms of readiness, reliability, and affordability.

23.3.4 Two-level Maintenance (TLM)

Maintenance of the weapon systems and mission support assets (i.e., materiel maintenance) is a critical element in the readiness and sustainability of combat forces. The distribution of maintenance workloads among the public and private sectors is instrumental in maintaining a robust and viable industrial base. Materiel maintenance operations support a wide range of weapon systems including ships, aircraft/helicopters, strategic missiles, and ground combat and tactical vehicles.

The legacy four-level maintenance includes direct support level, unit level, depot level, and a general support level. There are a few problems with the legacy four-level maintenance including that it (1) requires a large logistics footprint, (2) relies on removal of systems, (3) had built-in overhead burden, and (4) requires backup maintenance support. The purpose of TLM was to save money by reducing maintenance staffing, equipment, and base-level support without sacrificing force readiness. The TLM approach is to combine the direct-support-level and the unit-level maintenance to first-level maintenance and combine the depot-level and general-support-level to second-level support. The first-level maintenance is often called the field-level or organizational-level where the second-level maintenance is called the depot-level. The two maintenance components are distinguished largely by their relative capabilities, flexibility, agility, and capacity. Some of the benefits of a TLM are that it (1) reduces maintenance tiers, (2) minimizes duplication of work, (3) reduces the logistics footprint, (4) reduces procedural steps and people, and (5) increases flexibility and depth of capabilities.

23.3.4.1 Field-Level/Organizational Maintenance

The goal of the field-level maintenance is a quick turnaround to enhance operational availability. Field-level maintenance comprises shop-type work as well as on-equipment maintenance activities at maintenance levels other than depot. The field-level maintenance consists of the organizational level maintenance and the intermediate level activities.

This organizational-level maintenance is also known as the on-equipment maintenance. It is the responsibility of and performed by a using organization on its assigned equipment. The activities normally consist of inspecting, servicing, lubricating, and adjusting, as well as the replacing of parts or line replaceable unit with spare or serviceable assets taken from inventory minor assemblies and subassemblies. Typically the organizational-level maintenance is work performed in the field, on the flight-line, or at the equipment site, and is not only accomplished by maintenance personnel, but also by equipment operators. It is normally performed by an operating unit on a day-to-day basis to support operations of its assigned weapon systems and equipment. Organizational maintenance encompasses a number of categories, such as inspections, servicing, handling, preventive maintenance, and corrective maintenance.

The field-level maintenance repair effort for the ground segment is to establish spares for all ground segment hardware that is considered to be a line replaceable unit. The failed parts are then sent to the depot-level maintenance for further analysis and repair.

23.3.4.2 Depot-level Maintenance

The depot-level maintenance is performed in a specialized maintenance shop that is typically allocated to multiple operating units residing at a common operating location. This level of maintenance allows for a more thorough and time-consuming diagnostic testing and repair procedure, usually in support of failed items removed at the organizational-level of repair. Test equipment is commonly used at this level of repair to automate many test procedures. Depot-level maintenance entails materiel maintenance requiring the major repair, overhaul, or complete rebuilding of weapon systems, end items, parts, assemblies, and subassemblies; the manufacture of parts; technical assistance (hardware and software); and testing. Each military service manages and operates its own depot-level maintenance infrastructure.

For the ground segment, depot-level maintenance maintains a knowledge base for technical trouble shooting on return LRUs and establishes the necessary material for repair.

23.3.5 Performance-Based Logistics (PBL)

Performance-based logistics (PBL) is a support strategy that places primary emphasis on optimizing weapon system support to meet the needs of the warfighter. The PBL approach (a) documents the warfighter performance requirements as measurable metrics in the maintenance contract agreement, (b) appoints a single-point accountability for performance, and (c) develops support metrics and their accompanying incentives to ensure that performance objectives are met. In achieving reduction in the cost of logistics goals, the DOD has shifted the acquisition approach from buying parts, transactional goods, and services to buying performance. PBL defines the performance outcomes of weapon systems, ensures that responsibilities are assigned, provides incentives for achieving these performance goals, and facilitates the overall lifecycle management of system reliability, supportability, and total ownership costs. In performance-based contracts, a clear statement of outcomes to be achieved and the metrics needed to evaluate success are required.

The PSM has the responsibility for establishing the product support business model (PSBM) [7]. The PSBM describes the product support methodology to achieve an effective and efficient product. Product support considerations should begin prior to milestone (MS) A, with early requirements determination and continue through system design, development, operational use, retirement, and disposal. The PSBM defines the planning, development, implementation, management, and execution of product support over the sustainment lifecycle.

Executed through long-term, incentive-based contracts, PBL is a means of system sustainment that integrates supplier support and warfighter requirements with the objective of improving operational readiness while reducing logistics costs and reducing cycle time (e.g. repair cycle time, procurement cycle time).

PBL applies to new programs, capability and sustainment modifications, and re-procurement of systems, subsystems, and commodities.

23.3.6 Contractor Logistics Support (CLS)

Contractor logistics support is when a contractor, rather than the government, is responsible for the integration of logistics support functions such as providing engineering support; identifying requirements for spare and repair parts; maintaining facilities, materiel, and equipment; providing personnel; and performing maintenance on weapon systems. CLS is a method of obtaining support for a product throughout its lifecycle. Contractors can provide logistics support with a wide range of options, from interim contractor support, covering the initial fielding logistics support, to full contractor support. Temporary contractor support allows a service to defer investment in all or part of required support resources (spares, technical data, support equipment, training

equipment, etc.), while an organic support capability is phased in. CLS may be utilized as appropriate for all of the requisite logistics support, for specific logistics functions. or to support incremental deliveries of ground segment products.

23.3.7 Product Support Management

The PM is responsible for planning and executing the program to accomplish program objectives across the entire lifecycle, including the O&S phase. The PM is the designated individual with responsibility for and authority to accomplish program objectives for development, production, and sustainment to meet the user's operational needs. The PM is accountable for credible cost, schedule, and performance reporting to the MDA. A PM typically delegates the responsibilities for oversight and management of the product support function to the PSM.

The PSM is a key leadership position responsible in developing and implementing a viable product support strategy. The PSM needs to be able to interface effectively with senior leaders from other functional domains, including program management, contract management, business and financial management, and systems engineering and hardware and software engineers.

During acquisition, the program focus is primarily through the acquisition community with requirements input from the user and sustainment communities. Examples of these requirements include:

- Specification of design parameters for sustainment-related system performance capabilities
- Application of systems engineering to determine the right balance between the system's design requirements and the logistics support requirements to sustain the operational capabilities at an affordable price
- Application of the sustainment metrics to measure performance. Sustainment metrics include the KPP with their supporting KSAs
- The development of an integrated product support package to sustain the maintenance and support concepts that meet the materiel availability requirements
- Reduction of operating and support costs
- Identification and implementation of design changes to address evolving requirements, technological obsolescence, diminishing manufacturing sources, or materiel availability shortfalls

23.3.8 Sustaining Engineering

Sustaining engineering consists of a combination of systems engineering and product support life-cycle management strategies to achieve the desired sustainment metric outcomes for the program. These metrics include the DOD required KPP of Availability, the KSAs of Reliability and Total Ownership Cost, the recommended metric, Mean Down Time, plus other subordinate program metrics.

The *Product Support Manager Guidebook* [13] describes sustaining engineering as

> "those technical tasks (engineering and logistics investigations and analyses) to ensure continued operation and maintenance of a system with managed (i.e., known) risk. Sustaining Engineering involves the identification, review, assessment, and resolution of deficiencies throughout a system's lifecycle. It also involves implementation of selected corrective actions, to include configuration or maintenance processes, and the monitoring of key sustainment health metrics such as the following:
>
> - Collection and triage of all service use and maintenance data
> - Analysis of safety hazards, failure causes and effects, reliability and maintainability trends, and operational usage profile changes
> - Root cause analysis of in-service problems (including operational hazards, deficiency reports, parts obsolescence, corrosion effects, and reliability degradation)
> - The development of required design changes to resolve operational issues
> - Other activities necessary to ensure cost-effective support to achieve peacetime and wartime readiness and performance requirements over a system's life-cycle
>
> Sustaining engineering both returns a system to its baselined configuration and capability, and identifies opportunities for performance and capability enhancement. It includes the measurement, identification and verification of system technical and supportability deficiencies, associated root cause analyses, evaluation of the potential for deficiency correction and the development of a range of corrective action options. Typically business case analysis and/or lifecycle economic analysis are performed to determine the relative costs and risks associated with the implementation of various corrective action options. Sustaining Engineering also includes the implementation of

selected corrective actions to include configuration or maintenance processes and the monitoring of key sustainment health metrics."

23.3.9 Process and Discipline

23.3.9.1 Identifying and Evaluating Alternatives

Key activities during the MSA acquisition phase involve identifying and assessing alternatives and their system sustainment and product support implications. This process is critical because the resulting details will be used to guide the acquisition community in refining the concept selected and in identifying potential operating and support resource constraints. MSA is the first opportunity in acquisition to influence systems supportability by achieving system affordability via balancing technology opportunities with operational and sustainment requirements. This is the phase to consider requirement alternatives and is the phase that has the greatest bearing on the LCC. The emphasis is not only on the reliability and maintainability of potential materiel solutions, it is also on evaluating cost-effective responsiveness and the relevance of support system and supply chain alternatives.

MSA is also the phase to define the associated performance metrics to achieve the required effectiveness goals and the overall ability to accomplish a mission. The assessment includes the ability to sustain the system balancing between mission effectiveness, LCC, logistics footprint, and risk that best represents warfighter needs.

23.3.9.2 Sustainment Metrics

Metrics is a set of parameters or measures of quantitative assessment used for measurement, comparison or to track performance or production. Maintenance and sustainment program offices need insight to sustainment performance and work quality for all types of maintenance tasks. Metrics provide a means to quantitatively manage programs more effectively and to communicate and report the performance and quality of work up and down the channel.

The warfighters, or their operational user representatives, identify needed supportability and support-related performance capabilities parameters, e.g. sustainment metrics, footprint limitations, cost per operating hour, or diagnostic effectiveness. The performance-based metrics are sustainment enablers that establish the foundation for developing the sustainment KPP and supporting KSAs. The KPPs and KSAs are documented in the CDD and the CPD during the process of analyzing alternatives. The sustainment metrics are defined and analyzed against the alternatives along with a rough plan as to how they will be measured. The metrics are traceable to the initial capabilities document (ICD),

CDD, other JCIDS analysis, or agreement with the user community on the values for each metric and on documented analyses.

23.3.9.3 Alternative System Review (ASR)

One goal of the alternate system review (ASR) is to ensure the preferred system and product support solution satisfies the ICD. In general, this review assesses the alternative systems to ensure that at least one of the alternatives has the potential to meet the customers' needs and expectations, e.g. cost effective, affordable, operationally effective and suitable, and can be developed to provide a timely solution at an acceptable level of risk. The system concept is assessed by understanding the driving requirements for reliability, availability, maintainability, down time, lifecycle costs, and the enabling technologies required to meet user requirements. The sustainment requirements are proposed based on an analysis of alternatives that included cost, schedule, performance (including hardware, human, and software). Technology risks and the sustainment requirements are to be consistent with technology maturity, the proposed program cost, the schedule for the technical baseline, and preferred support concept. The support concept will include the conceptual description, scope, and risk for both the system and any supply chain system and software needs beyond what is currently available.

23.3.9.4 Lifecycle Sustainment Plan Informing Informing Request for Proposals

The LCSP, a government document, is the key communication tool that the government uses in their requests for proposals (RFPs) to potential bidders [10]. The LCSP is used by the government to convey the baseline product support strategy, sustainment performance requirements, government organizational structure, regulatory/statutory requirements, and a high-level schedule. The contractor proposal is expected to include their approach to accomplish contract requirements, the design-to requirements including verification method, and alternative strategy for affordable requirements. An example outline for an LCSP is:

1 Introduction
2 Product support performance
  2.1 Sustainment performance requirements
  2.2 Demonstrated (tested) sustainment performance
3 Product support strategy
  3.1 Sustainment strategy considerations
  3.2 Sustainment relationships

4 Product support arrangements
  4.1 Contracts
  4.2 Performance-based agreements
5 Product support package status
  5.1 Program review results
  5.2 Product support package assessment
6 Regulatory/statutory requirements that influence sustainment performance
7 Integrated Schedule
8 Funding
9 Management
  9.1 Organization
    9.1.1 Government program office organization
    9.1.2 Program office product support staffing levels
    9.1.3 Contractor(s) program office organization
    9.1.4 Product support team organization
  9.2 Management approach
    9.2.1 Product support manager roles and responsibilities
    9.2.2 Sustainment risk management
10 Supportability analysis
  10.1 Design interface
    10.1.1 Design analysis
    10.1.2 Technical reviews
  10.2 Product support element determination
  10.3 Sustaining engineering
11 Additional sustainment planning factors

LCSP annexes (depending on program phase/service requirements):
- Product support business case analysis
- System disposal plan
- Preservation and storage of unique tooling
- Core logistics analysis
- Source of repair analysis
- Service-specific requirements

23.3.10 Acquisition Phase

Product support capability applies across all stages of the DOD acquisition process. During the pre-systems acquisition and systems acquisition stages, it is the capability being developed by the development contractor for use during the sustainment stage to perform product and software product support. During the sustainment stage itself, it is the capability used by support personnel to perform product support.

Subparagraphs of DODI 5000.02 describe the major program lifecycle phases, the activities and deliverables associated with each phase, the expected maturity

level. The major program lifecycle phases are MSA, TMRR, Engineering and Manufacturing, P&D, and O&S phases. Refer to DODI 5000.02, *PSM Guideline*, and *Defense Acquisition Guidebook (DAG)* for details [1, 2, 13].

23.3.10.1 Materiel Solution Analysis Phase

The purpose of the materiel solution analysis phase is to conduct analysis and other activities needed to choose the concept for the product that will be acquired, to begin translating validated capability gaps into system-specific requirements including the KPPs and KSAs, and to conduct planning to decide on the acquisition strategy for the product. Analysis of alternatives (AoA) solutions, key trades among cost, schedule, and performance, affordability analysis, risk analysis, and planning are all considerations.

For product support, this phase defines initial supportability objectives and a sustainment strategy and evaluates technology ability for implementation. Sustainment strategy considerations are the degree to which a system's design and planned logistics resources support system readiness requirements and wartime utilization, resources to facilitate the detection, isolation, and timely repair/replacement of system anomalies, items necessary for system operation, real world constraints, and the operational environment (e.g., personnel, equipment, technical support data, and maintenance procedures).

During this phase the PSM will establish notional maintenance concepts and metrics, identify user needs and environmental constraints impacting sustainment, identify enabling technologies and the test and evaluation strategy for verification of KPP and KSAs, and estimate LCC drivers.

23.3.10.2 Technology Maturation and Risk Reduction Phase

The purpose of the technology maturation and risk reduction phase is to reduce technology, engineering, integration, and lifecycle cost risk to the point that a decision to contract for engineering and manufacturing development (EMD) can be made with confidence in successful program execution for development, production, and sustainment.

For product support, this is the phase to refine sustainment objectives and requirements, set sustainment metrics and requirements, define the supportability design features required to achieve KPPs and KSAs, identify the design-in sustainment features, and establish product support package requirements and design the PSP.

During this phase, the PSM will establish sustainment concept and execution plan framework, set sustainment metrics goals and/or thresholds and test and evaluate an approach to verify the related design features and KPP and KSA.

The sustainment contracting strategy, including the extent performance-based logistic contracts, is documented in the acquisition strategy. The LCSP, including identification of logistics risks and associated risk mitigation strategies, is written and approved in preliminary design review (PDR). The LCSP will include a preliminary support strategy leveraging a best value mix of public (organic), and private (contractor) support.

23.3.10.3 Engineering and Manufacturing Development Phase

The purpose of the engineering and manufacturing development phase is to develop, build, and test the acquired system to verify that all operational and derived requirements have been met, and to support production or deployment decisions.

For product support, this is the phase to define the product support structure and product support package requirements, establish sustainment metrics verification methods, and to complete a fielding plan and development details.

At this phase, the sustainment and product support planning is complete; the sustainment strategy, mix of public and private partnership, and their roles and responsibilities have been identified. The sustainment and product support capabilities and the associated logistics processes and products are tested and demonstrated, the supply chain performance validated, and budget requirements are adjusted based on the design and test results. The LCSP is updated with product support package and supply chain information including detailed product support element requirements, details about the product support package development and implementation, performance verification methods, and fielding plans.

23.3.10.4 Production and Deployment Phase

The purpose of the production and deployment phase is to produce and deliver requirements-compliant products to receiving military organizations. In this phase, the product is produced and fielded for use by operational units.

For product support, this is the phase to prove through successful tests and demonstrations (in an operational environment) that the product support package and sustainment and product support capabilities, including associated logistics processes and products, can be fielded to support an operational environment. It includes performance measured against availability, reliability and cost metrics, and support systems and services delivered to each category of operational site. Plans are developed and implemented to address issues based on performance data.

During this phase, the product support package elements are refined. The site fielding plans are refined with details. The LCSP is updated with refined product support package elements, fielding plan details, and adjustments and a logistics assessment. The update includes details of how the sustainment performance will be measured, managed, assessed, and reported. Cost drivers and availability degraders are identified.

23.3.10.5 Operations and Support (O&S) Phase

The purpose of the O&S phase is to execute the product support strategy, satisfy materiel readiness and operational support performance requirements, and sustain the system over its lifecycle including disposal. The O&S phase begins after the production or deployment decision, and is based on a milestone decision-authority-approved LCSP.

This is the maintenance and sustainment phase where the product support package is fully in place, including depot repair capability. The support systems and services are delivered and fully integrated into the operational environment.

During this phase, depot maintenance is performed. The processes per the approved LCSP are followed, e.g. regularly measure the sustainment and product support performance metrics with corrective action taken, plan product improvement, modifications, upgrades, refine the support strategy leveraging the best value mix of public and private logistics processes, services and products, and implement equipment retirement and disposal as required.

The LCSP is updated with adjusted fielding plans, sustaining engineering processes and continuous improvement plans, logistics assessments and updated on how the system and supply chain are performing, and program or funding changes are adjusted as required.

23.4 Practices

23.4.1 Core Activities

There are several important program management activities for programs entering into the maintenance and sustainment phase [14]. Program managers are required to “develop and implement performance-based logistics strategies that optimize total system availability while minimizing cost and logistics footprint [13].” DODD 5000.01 requires that “Planning for Operation and Support and the estimation of total ownership costs shall begin as early as possible. Supportability, a key component of performance, shall be considered throughout the system lifecycle.” DOD Directive 5000.01, requires program managers to “develop and implement performance-based product support strategies that optimize total system availability while minimizing cost and

logistics footprint. Sustainment strategies shall include the best use of public and private sector capabilities through government/industry partnering initiatives, in accordance with statutory requirements [13]."

The PM, as the lifecycle manager, is responsible for accomplishing program objectives across the lifecycle, including the O&S phase. The PM, along with the PSM and the life-cycle logisticians, is responsible for influencing the design and providing effective, timely product support capability to achieve the system's materiel readiness and sustain operational capability.

23.4.1.1 Lifecycle Sustainment Implementation

Lifecycle sustainment involves the early planning, development, implementation, and management of a comprehensive, affordable, effective performance-driven logistics support strategy, ensuring sustainment considerations are integrated into all planning, implementation, management, and oversight activities associated with the acquisition, development, production, fielding, support, and disposal of a system across its lifecycle. Activities include:

- Participating in the design process to acquire a highly supportable and sustainable system.
- Providing affordable, reliable, effective support strategies and systems that meet the user's requirements with optimum materiel availability.
- Developing the appropriate metrics to validate and verify the system engineering design process, and measure the performance of the support strategy and supply chain.
- Providing the user effective systems with the minimal logistics footprint (e.g., the measurable size or "presence" of logistics support, including manpower, required to deploy, sustain, and move a system).
- Developing more integrated and streamlined acquisition and statutorily compliant logistics support processes.
- Facilitating iterative technology enhancements during the system lifecycle.

23.4.1.2 Product Support Implementation

Product support is the application of the package of integrated product support elements and support functions necessary to sustain the readiness and operational capability of the system. During the acquisition process the focus is on influencing the design for supportability and fielding the support concept to satisfy user-specified requirements for sustaining system performance at the lowest LCC. Considerations include:

- Availability of support to meet warfighter-specified levels of combat and peacetime performance.
- Logistics support that sustains both short- and long-term readiness.
- Management of LCC through analysis and decision prioritization.
- Maintenance concepts to integrate the product support elements and to optimize readiness while drawing upon both public and private sources.
- Data management and configuration management that facilitates cost-effective product support throughout the system lifecycle.
- A diminishing manufacturing-sources and material-shortages management process that ensures effective, affordable, and operationally reliable systems.
- Operator and maintainer training to encompass the full capability of the system.

23.4.1.3 Sustainment Metrics Implementation

In a performance-based environment, sustainment related requirements, with a specified range of minimum mandatory (threshold) and target (objective) performance capability design parameters, are established with accompanying metrics covering the entire enterprise. This includes the system and the supply chain supporting it. The metrics key attributes are (1) traceable to user requirements, (2) achievable and verifiable, and (3) minimum reporting.

The specific metrics are tailored to the program and the operational and sustainment needs. At a minimum, the metrics consist of an outcome metric meaningful to the user in achieving and sustaining the operating tempo, a materiel metric to measure the system's quality, a response metric to measure the quality of the logistics system, and a cost metric.

23.5 Key Lessons Learned

There are several key lessons learned during the course of acquisition, request for proposal, source selection, design and implementation, system maintenance and sustainment. These are watch items for programs to avoid when transiting into the maintenance and sustainment phase.

- Plan for maintenance and support early in the acquisition cycle
- Ensure that the system is well documented
- Limit dependence on contractors by using extensible, easy-to-use products
- Government does not have the proper data rights

  Contracts may not have proper language for government's rights in technical data and computer software. The government's lack of access

to proprietary technical data results in dependence on specific contractors for expertise and can limit and even preclude the possibility of competition. There are many Government Accountability Office (GAO) reports which address this issue, e.g. GAO-06-839 of July 2006, GAO-10-833 of July 2010, and GAO-13-325 of March 2013 [15–17].

- Insufficient DMS management

  Diminishing manufacturing sources and material shortage is when the source, e.g. manufacturers or suppliers of items, raw materials, or software, discontinues providing the items for the ongoing production capability or lifecycle support of a weapons system or when there are shortages in any training, support, or test equipment already in the field. The risk is unavoidable and can endanger mission effectiveness.

- Weak configuration management

  System changes, (e.g., hardware, software, configuration), made as temporary patches or quick fixes did not follow a formal configuration change control process. The risk is that the baseline at various operational sites will be unsynchronized.

- Inadequate product support baseline

  Inadequate test procedures, technical orders, training material, or product support baseline does not perform as expected in the operational environment with operational data and databases and does not meet user's intent.

- Inadequate product support environment

  Product support environment(s) (e.g., hardware, software, tools) do not provide an adequate capability for defect or enhancement analysis, fixing, and/or verification.

- Inexperienced staff

  Sustaining engineers do not have the adequate skill and experience to perform assigned tasks

23.6 Government and Contractor Enabling Processes and Products

Key program acquisition documents should include sustainment aspects that cut across lifecycle phases.

23.6.1 ICD/CDD/CPD

These documents specify authoritative and testable performance capabilities for the program. The ICD prefaces a system materiel decision and evolves into the CDD, which prioritizes KPP and subset KSA performance capability design and development parameters. The baseline CPD is finalized after the system-level critical design review and before Milestone C.

23.6.2 Analysis of Alternatives (AoA)

The AoA describes and includes the results of the supportability analyses and trade-offs conducted to determine the optimum support concept as part of the preferred system concept. It should also include the assumptions used in the analyses.

23.6.3 Technology Development Strategy

The technology development strategy (TDS) includes the specific new sustainment-related technologies required to achieve the sustainment KPP/KSAs, the technologies required to achieve logistics performance over what is currently achieved in today's operational environment.

23.6.4 Acquisition Performance Baseline

The acquisition performance baseline (APB) documents the performance requirements, schedules, and program cost funding and estimates. It includes sustainment KPPs and KSAs, measurement metrics, and all programmatic direction affecting lifecycle support strategy planning and execution.

23.6.5 Acquisition Strategy

Acquisition strategy is a business and technical management approach designed to achieve program objectives within the resource constraints imposed. It is the framework for planning, directing, contracting for, and managing a program. It provides a master schedule for research, development, test, production, fielding, modification, post-production management, and other activities essential for program success. The acquisition strategy is the basis for formulating functional

plans and strategies (e.g., test and evaluation master plan, acquisition plan, competition, systems engineering plan, etc.).

The acquisition strategy describes the PM's approach for acquiring the system and its support. This includes the acquisition strategy for achieving the sustainment metrics and acquiring the PSP. An acquisition strategy includes the key upcoming contracting actions and the timeline to acquire the product support elements necessary to maintain the system's readiness and operational capability. The strategy addresses how the product support package is required to support the materiel management, distribution, technical data management, support equipment, maintenance, training, configuration management, engineering support, supply support, and failure reporting/analysis, and functions. It should summarize the approach for acquiring key enablers for achieving the sustainment metrics, e.g., using diagnostics, prognostics, a modular open systems approach, and supporting reliability growth.

23.6.6 Test and Evaluation Master Plan

The test and evaluation master plan (TEMP) is critical to achieve sustainment metrics thresholds and objectives. It includes a description of the requirements, test points, and methods for each of metric as well as any appropriate enabler or logistics consideration.

23.6.7 Systems Engineering Plan

The systems engineering plan (SEP) approach is an integral part in designing for sustainment and supporting the design. This plan includes the integration of sustainment metrics with other requirements. Sustainment aspects should be included in specialty engineering as follows:

- The human systems integration (HSI) plan includes maintenance, sustainment, and other support personnel aspects
- Integrated product teams (IPTs) includes sustainment
- The development and update of the failure mode, effects and criticality analysis (FMECA) matrix; identification of critical safety items (CSIs); failure reporting, analysis, and corrective action system (FRACAS); and trend analysis for maturation purposes of the system and its support system
- Technical baselines (functional, allocated, and product) address the end item system and its product support package elements
- Technical reviews includes sustainment and product support package technical maturity against the baselines

23.6.8 Diminishing Manufacturing Sources/Materiel Shortages (DMSMS) Plan

The DMSMS management process includes proactively identifying and mitigating DMSMS issues that affect their availability and supportability throughout the entire life of the program.

23.6.9 Sustainment Quad Chart

The sustainment quad chart provides sustainment information in a standardized format use in reporting status at overarching integrated product team and DAB reviews. It is used to strengthen sustainment governance by providing senior management visibility of key sustainment factors to help ensure the program manager's sustainment strategy meets the warfighter materiel readiness and long-term affordability objectives.

23.6.10 Lifecycle Sustainment Plan

The LCSP addresses a program's product support strategy for accomplishing the supportability objectives across the lifecycle, including during the O&S phase. The LCSP is the key logistics acquisition deliverable required in the defense acquisition system.

23.7 References

1. *Operation of the Defense Acquisition System*. Department of Defense Instruction Number 5000.02 (DODI 5000.02), January 15, 2015. http://www.dtic.mil/whs/directives/corres/pdf/500002p.pdf

2. Defense Acquisition Guidebook (DAG). https://dag.dau.mil/Pages/Default.aspx

3. *Glossary: Defense Acquisition Acronyms and Terms*, *16th Edition*. September 2015. http://www.dau.mil/pubscats/Pages/preface.aspx

4. USD AT&L Policy Memo "Strengthened Sustainment Governance for Acquisition Program Reviews" ("Sustainment Quad Chart"). Dated April 5, 2010. https://acc.dau.mil/CommunityBrowser.aspx?id=360876&lang=en-US

5. *Manual for the Operation of the Joint Capabilities Integration and Development System (JCIDS)*. JCIDS Manual. February 2015. https://acc.dau.mil/jcids

6. CJCSI 3170.01 I. January 28, 2015.
https://acc.dau.mil/jcids

7. *Acquisition/Logistics: Integrated Life Cycle Management.* Air Force Instruction 63-101/20-101, March 07, 2013.
www.e-publishing.af.mil

8. *Integrated Life Cycle Management*. Air Force Pamphlet (AFPAM) 63-128. July 10, 2014.

9. *Integrated Life Cycle Management*. Air Force Policy Directive 63-1/ 20-1. July 3, 2012.

10. *Life Cycle Sustainment Plan Sample Outline*. September 2011.
http://www.acq.osd.mil/se/docs/LCSP-Sample-Outline-10Aug2011.pdf
https://acc.dau.mil/lcsp-outline

11. *Integrated Product Support Element Guidebook*. December 2011.
https://acc.dau.mil/CommunityBrowser.aspx?id=495014

12. *Product Support Manager Guidebook*. U.S. Department of Defense. April 2011.

13. *The Defense Acquisition System*. Department of Defense Directive Number 5000.01 (DoDD 5000.01), May 12, 2003. Certified Current as of November 20, 2007
http://www.dtic.mil/whs/directives/corres/pdf/500001p.pdf

14. *Weapons Acquisition: DOD Should Strengthen Policies for Assessing Technical Data Needs to Support Weapons Systems*. Government Accountability Office (GAO) report GAO-06-839. July 2006.
http://www.gao.gov/new.items/d06839.pdf

15. *Federal Contracting: Opportunities Exist to Increase Competition and Assess Reasons When Only One Offer is Received*. Government Accountability Office (GAO) report GAO-10-833. July 2010.
http://www.gao.gov/new.items/d10833.pdf

16. *Defense Contracting: Actions Needed to Increase Competition.* Government Accountability Office (GAO) report GAO-13-325. March 2013.
http://www.gao.gov/assets/660/653404.pdf

17. *Principal Deputy Under Secretary of Defense (PDUSD)*. Life-Cycle Sustainment Plan, document streamlining, memorandum. September 14, 2011. https://dap.dau.mil/policy/Lists/Policy%20Documents/Attachments/3303/USA005157-11_SignedLCSPMemo_14Sep2011.pdf

23.8 Bibliography

Maintenance Policy & Programs (Office of the Assistant Secretary of Defense for Logistics & Material Readiness)
http://www.acq.osd.mil/log/mpp/index.html

Sustaining Engineering ACQuipedia
https://acc.dau.mil/CommunityBrowser.aspx?id=514837

Acquiring and Enforcing the Government's Rights in Technical Data and Computer Software Under Department of Defense Contracts, Seventh_Edition, Space and Missile Systems Center, El Segundo, CA. August 2015.

Donahue, Charles D. and B. McKinzey. *Configuration Management*, TOR-2006(8583)-1, The Aerospace Corporation, El Segundo, CA, August 2005.

Shaw, Brian E. *Systems Engineering Requirements and Products*. TOR-2005(8583)-3, Rev. B, The Aerospace Corporation, El Segundo, CA. April 15, 2010.

Adams, Richard I. and Suellen Eslinger. *Software Sustainment Guidance*. TOR-2013-00693, The Aerospace Corporation, El Segundo, CA. December 31, 2013.

Houston, Daniel X., Nancy S. Kern, Karen R. Sharp, Bonnie R. Troup, and Jean C. Wang. *Recommended Software Measures for MILSATCOM Sustainment Programs*. TOR-2013-00750, The Aerospace Corporation, El Segundo, CA. September 27, 2013.

23.9 Acronyms

ACAT	acquisition category
AF	Air Force
AFI	Air Force Instruction
AFPD	Air Force Policy Directive
AoA	analysis of alternatives
AP	acquisition plan
APB	acquisition program baseline
ASR	alternative system review

AT&L	acquisition, technology, and logistics
CDD	capability development document
CDR	critical design review
CI	configuration item
CLS	contractor logistics support
CPD	capability production document
CSI	critical safety items
DAB	defense acquisition board
DAG	Defense Acquisition Guidebook
DMSMS	diminishing manufacturing sources/materiel shortages
DOD	Department of Defense
DODD	Department of Defense Directive
DODI	Department of Defense Instruction
EMD	engineering and manufacturing development
FMECA	failure mode, effects, and criticality analysis
FRACAS	failure reporting, analysis, and corrective action system
FRP	full rate production
GAO	Government Accountability Office
HIS	human systems integration
ICD	initial capabilities document
ICS	interim contractor support
ILCM	integrated lifecycle management
ILS	integrated logic support
IOC	interim operational capability
IPS	integrated product support
IPT	integrated product teams
JCIDS	Joint Capabilities Integration and Development System
KPP	key performance parameters
KSA	key system attributes
LCC	lifecycle cost
LCSP	lifecycle sustainment plan
MDA	milestone decision authority
MOP	measures of performance
MS	milestone
MSA	material solution analysis
O&S	operations and support
P&D	production and deployment
PBL	performance based logistics
PCA	physical consideration audit
PDR	preliminary design review
PHS&T	packing, handling, storage, and transportation
PM	program manager
PSBM	product support business model
PSM	product support manager
R&D	research and development

RFP	request for proposal
SMC	Space and Missile Systems Center
TDS	technology development strategy
TLM	two-level maintenance
TMRR	technology maturation and risk reduction
TO	technical orders

Chapter 24
Technology Refresh: Updating Ground Elements

Mel M. Cutler
Computer Applications and Assurance Subdivision
Computers and Software Division
Martha I. Johnson
Software Acquisition and Modeling Department
Software Engineering Subdivision

24.1 Introduction/Background

Declining budgets are forcing systems, including ground systems, to perform well past their planned operational life. With this comes interoperability and obsolescence challenges with hardware and software. Historically, the approach to resolving obsolescence issues, or managing technology changes, would be to redesign and integrate new items as the issues arise, or to procure large quantities of parts to meet the projected parts demand. This approach is time consuming and costly. In addition, because of long acquisition timelines, by the time the new design is ready, it may already be obsolete.

The term "technology refresh" is commonly used for programs. However, there are no government policies, regulations, or instructions specifically using the phrase "technology refresh." The term technology refresh is interpreted as making a change that incorporates the latest version of an existing hardware or software technology to correct an existing problem or to avoid obsolescence during the life of the product.

Technology refresh ensures continued supportability throughout the product's life. It includes the periodic replacement of hardware and software components such as:

- Non-developmental items (NDI)
  - Commercial-off-the-shelf (COTS)
  - Government off the shelf (GOTS)
  - Reuse software
  - Free and open source software (FOSS)
- Processors
- Displays
- Operating systems (OSs) and other system software
- Radio frequency (RF) and digital hardware components

The development of a planned and organized technology refresh program is critical to ensure long-term system availability. A technology refresh program should have an enterprise perspective and include many functional areas, such as: engineering, supply chain management, obsolescence, diminishing manufacturing sources and material shortages, capability enhancement, lifecycle sustainment planning, cybersecurity, and metrics to guide and drive resources and efforts.

Effective implementation of technology refresh necessitates managing the appropriate balance between system performance, affordability, and availability. To prevent technology obsolescence issues from directly affecting the system performance, it must be planned and implemented throughout the acquisition lifecycle to ensure the system can be maintained for as long as it is needed.

Increasingly, cloud service providers (e.g., Amazon) are looking to address this problem on behalf of even large-scale users of computing facilities by providing a "converged" architecture where the user can host middleware and applications on an outsourced infrastructure that is continually refreshed, from both a hardware and software standpoint, at a scale that provides economies to users. The cost of acquiring, integrating, testing, and deploying new configurations is amortized over a scaled-up user base and can be much less costly and lower risk than if it were managed by the guest organizations on their own. Nonetheless, the challenges at integration levels above the infrastructure level remain as they are not subject to the same economies of scale and are driven by the specific needs of individual programs.

The remainder of this chapter covers the following topics: relevant definitions, governing or reference documents related to the process of technology refresh, a reference software architecture to provide the context for software technology refresh, a trade space defining the range of technology refresh options, and an exposition of governing principles and program-specific considerations that would tend to drive the system maintainer towards one portion of the trade space or another, with specific examples.

24.2 Definitions

Commercially available-off-the-shelf item (COTS) A commercial item (CI) sold in substantial quantities in the commercial marketplace and offered to the government under a contract or subcontract at any tier, without modification, in the same form in which it was sold in the marketplace. This definition does not include bulk cargo such as agricultural products or petroleum [1].

Diminishing Manufacturing Sources and Material Shortages (DMSMS) The loss, or impending loss, of manufacturers of items or suppliers of items or of raw materials. This can be caused by many factors including new or evolving

science, detection limits, toxicity values, and regulations related to chemicals and materials resulting in significant impact on DOD's supply chain and industrial base (IB). This situation may cause shortages that endanger the lifecycle support and capability of the weapon system or equipment [2]. Note: per *Integrated Lifecycle Management* DMSMS includes items, raw materials, or software [3].

Government-off-the-shelf item (GOTS) A term for software and hardware government products that are ready to use. They were created and are owned by the government. Typically GOTS are developed by the technical staff of the government agency for which it is created. It is sometimes developed by an external entity, but with funding and specification from the agency. Because agencies can directly control all aspects of GOTS products, these are sometimes preferred for government purposes. GOTS software solutions can normally be shared among government agencies without additional cost. GOTS hardware solutions are typically provided at cost (i.e., R&D costs not recouped).

Interoperability The ability of systems, units, or forces to provide data, information, materiel, and services to, and accept the same from, other systems, units, or forces, and to use the data, information, materiel, and services exchanged to enable them to operate effectively together. IT interoperability includes both the technical exchange of information and the end-to-end operational effectiveness of that exchange of information as required for mission accomplishment. Interoperability is more than just information exchange. It includes systems, processes, procedures, organizations, and missions over the lifecycle and must be balanced with cybersecurity (formerly Information Assurance) [4].

Technology refresh Making a change that incorporates a new version of an existing hardware or software technology to correct an existing problem or to avoid obsolescence during the life of the product.

24.3 Objectives

The objective of technology refresh is to provide the timely insertion of technology in the ground elements, in the quality and quantity necessary to meet the needs of the program or set of programs it supports at the "best value". This requires technology refresh to be integrated into the full acquisition lifecycle activities to ensure that the resulting technology:

- Will meet performance and material readiness requirements
- Will be affordable, resilient, and responsive to changing customer needs across the lifecycle

This chapter is intended to be a source of information for the ground elements, providing a mapping as to when technology refresh related activities should be

planned and addressed within the acquisition and/or program lifecycle. The major points of interface with the existing systems engineering and sustainment engineering processes will also be addressed.

24.4 Practices

As identified in the Introduction, there are no DOD policies, regulations or instructions specifically addressing technology refresh or using the phrase "technology refresh". However, a number of directives and instructions that include requirements and guidance for technology refresh:

Directive	Requirements/Guidance Summary
Public Law 106-398 [5]	Requires the redesign of processes that the system supports to reduce costs, improve effectiveness and maximize the use of COTS technology.
OMB M-04-04 [6]	OMB guidance outlines a 5 step process by which agencies should meet their e-authentication assurance requirements. Step 5 requires: "Periodically reassess the information system to determine technology refresh requirements. The agency must periodically reassess the information system to ensure that the identity authentication requirements continue to be valid as a result of technology changes or changes to the agency's business processes. Annual information security assessment requirements provide an excellent opportunity for this. Agencies may adjust the identity credential's level of assurance using additional risk mitigation measures. Easing identity credential assurance level requirements may increase the size of the enabled customer pool, but agencies must ensure that this does not corrupt the system's choice of the appropriate assurance level."
DOD 5000.01 [7]	E1.1.17 requires "Program managers shall develop and implement performance-based logistics strategies that optimize total system availability while minimizing cost and logistics footprint. Trade-off decisions involving cost, useful service, and effectiveness shall consider corrosion prevention and mitigation. Sustainment strategies shall include the best use of public and private sector capabilities through government/industry partnering initiatives, in accordance with statutory requirements." E1.1.27 requires "Acquisition programs shall be managed through the application of a systems engineering approach that optimizes total system performance and minimizes total ownership costs. A modular, open-systems approach shall be employed, where feasible."

Directive	Requirements/Guidance Summary
	E1.1.29 requires "…Supportability, a key component of performance, shall be considered throughout the system lifecycle."
DODI 5000.02, Operation of the Defense Acquisition System [8]	Includes requirements that extend throughout the acquisition lifecycle for continuous analysis and planning to address total lifecycle cost, interoperability, sustainability requirements and processes, and operational supportability
DODI 8330.01, Interoperability of Information Technology (IT), Including National Security Systems (NSS) [4]	Establishes the policy and requirements for IT interoperability including development, test, certification and prerequisites for connection of IT.
AFI 63-101/20-101, Integrated Life Cycle Management [3]	Requires systems engineering to manage the system development and sustainment addressing each system (hardware, software, and human) and to develop design requirements to apply open systems (modular open systems approach [MOSA]) and open technology development that allow components to be added, modified, replaced, removed and/or supported by different vendors throughout the system's lifecycle. In addition it identifies that key software focus areas to be addressed throughout the lifecycle and be incorporated in the program systems engineering plan (SEP).
NIST 800-63-2 [9]	This technical guidelines supplement OMB M-04-04 guidance, *E-Authentication Guidance for Federal Agencies* that requires periodic reassessment of the information system to determine technology refresh requirements.
NIST SP 800-37, Revision 1 [10]	Provides guidelines on the frequency, depth, and breadth of periodic technology refresh reassessments.
DMSMS Acquisition Guidelines [11]	Provides guidance on the acquisition and sustainment of electronic components and includes as a part of this guidance that the statement of work should contain language and evaluation criteria addressing the plan for periodic replacement of components (i.e., technology insertion of technology refresh).
IEEE Guide – Adoption of ISO/IEC TR 24748-3:2011 [12]	Contains numerous informative notes regarding obsolescence and technology refresh that are embedded in the ISO/IEC 12207:2008 and ISO/IEC 15288 standards. Although not requirements, these notes are intended to clarify the intent of the activities.

24.5 Planning for Technology Refresh

Planning for technology refresh is an essential initial step needed to ensure long-term system availability. Planning and managing technology refreshment from the program initiation and throughout the product lifecycle will help mitigate risks that could significantly impact both hardware and software component operations, maintenance, support, design and certifications. The "*Manager's Guide to Technology Transition in an Evolutionary Acquisition Environment*" [13] provides the following advice to the acquisition, R&D and sustainment communities: "No matter whether your system uses defense-unique technology or commercial technology—particularly in the electronics and computer components pervasive to many weapons systems—changes and obsolescence will be continual. The way to deal with these changes and obsolescence is to design for them, plan for, budget for, and have technology refreshment programs in place so improvements in both capability and affordability can be incorporated throughout the useful life of the system."

Unplanned technology refresh may represent a disruption to the system performance and readiness at an inopportune time. For example, there may be unscheduled downtime needed to incorporate the changes impacting the mission performance. Mission requirements may have to be deferred until solutions to the problem are found. In addition, technology refresh often is not budgeted and the resources (e.g., personnel, facilities, equipment) are not available to perform the refresh.

By integrating key aspects of technology refresh into the ongoing program activities, many of these risks and issues may be avoided. Figure 24-1 illustrates a program's system lifecycle stages and some of the key engineering activities that should be performed to mitigate technology risks to the program throughout the lifecycle. It is important to note that technology refresh does not just take place after production, but can start as early as the concept development stage to the end of product support. As a result, planning for refresh needs to be defined for each successive program increment from the start of the system lifecycle.

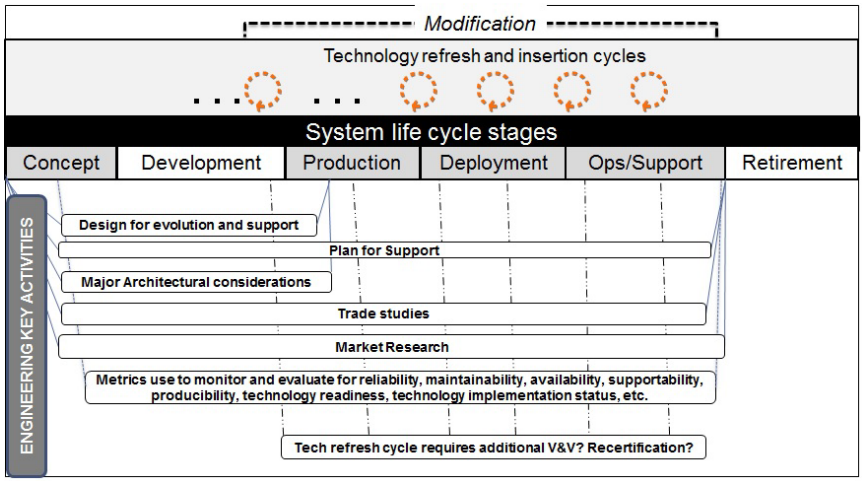


Figure 24-1. Key technology refresh systems engineering activities.

24.5.1 Concept Development

During the concept development stage, the initial concept and acquisition strategy are defined, technology prototypes are being developed and trade studies decisions are made balancing many attributes that affect the full lifecycle. The emphasis is on innovation and competition and on drawing from existing sources from a wide range of sources while balancing lifecycle affordability in the engineering decisions that include planning performance-based logistics. Planning for technology refresh should be reflected in the initial concept, acquisition strategy, and the SEP.

Planning for support of the system with integrated technology refreshment should describe the approach to full lifecycle management of obsolescence, technology refreshment implementation, and maintenance of the hardware and software. It should include the planned effort for sustainment and administration, customer support, and the help desk needed to support and manage these efforts. In planning for technology refresh, the following should be considered and defined for each technology item:

- What is the minimum operational life threshold(s) of a system which, if below that threshold the system performance is questionable or compromised?
- Should the system have an operational life beyond the threshold; is it possible to support?

24.5.2 Development stage

The development stage focuses on reducing technology risk and selecting the appropriate set of technologies that will become the most operationally useful.

Architecture considerations are a major factor in these decisions to ensure that downstream refresh activities can be performed. Open system architectures designed for interoperability allow for ease of integration, portability, scalability and technology refresh. Other benefits to open system architectures are that a program is not usually locked into proprietary technology solutions, a single supply source for the life of the program, and it mitigates risks of technology obsolescence. The open systems approach is identified in the *Defense Acquisition Guidebook* [14] as a best practice. One architectural approach to this is outlined in section 24.6.1

During the development of the request for proposal (RFP), statement of work, and resulting contract, requirements that support technology refresh need to be defined. These requirements should address:

- Open system architecture requirements
- Appropriate use of COTS, GOTS, and NDI
- Contractor delivery of technology insertion plans that include the plans to refresh COTS (hardware and software) for each developed increment and final delivery throughout the period of performance
- Lifecycle design requirement decisions that continuously focus on item reliability, maintainability, supportability, producibility, cost, and viable technology improvement options.
- Planning for refresh cycles throughout the system lifecycle and the resources (personnel, equipment, facilities,…) to support the cycles
- System lifecycle obsolescence management (both software and hardware)
- Decision gates at defined milestones which include evaluation of whether technology should be refreshed, inserted, or retired
- The contractor's include the plans for technology refresh in their:
  - Integrated master plan (IMP)
  - Integrated master schedule
  - Systems engineering management plan (SEMP)
  - Software development plan (SDP)
  - Software transition plan (STP)
  - Life cycle management plan (LCMP)
  - Life cycle sustainment plan (LCSP)
  - Risk management plan
- Licensing and data rights of any technology that is used in the system

- Inclusion of a clause that requires that, at the time of delivery, the contractor shall ensure that all software and hardware technology is supported by an active vendor.

Evaluation of the offerors proposed approach and proposed architectures should be performed to determine that the planned approach is "designed for evolution and support" and that it will support interoperability, minimizing risk and reducing overall lifecycle costs. Aerospace ATR-2012(9010-12), "*Evaluating Software Architectures in Space and Ground Systems*" [15] provides a framework for evaluating National Security Space (NSS)-related programs' software architecture. Included in this report, are evaluation criteria to determine the suitability of COTS and GOTS products for the system. COTS or GOTS can introduce architecture-related constraints on the system, as well as lead to obsolescence if not continuously monitored. In addition, TOR-2011(8506)-117, "*Integrating Software Topics into the Request for Proposal,*" [16], provides evaluation criteria for software architecture, rights in technical data, and software.

After contract award, metrics should be analyzed throughout development to:

- Monitor and evaluate technical items for reliability, maintainability, supportability, producibility, and impact on overall total costs.
- Evaluate the overall technology readiness of the system, the implemented technology status, and performance to technology refresh implementations
- Monitor risks related to technology refreshment-related risks, including monitoring suppliers.

24.5.3 Production, Deployment and Operations/Support

During production, deployment, and operational support stages, the continual monitoring of technology is performed using the same metrics used in earlier stages. Throughout these stages engineering continues to monitor and evaluate technical items for reliability, maintainability, supportability, producibility and impact on overall total costs. Suboptimal performance, or non-performance may be indicators that new technology may be needed. In addition, technology refresh may be required to comply with updated certification requirements (e.g., cybersecurity), revised environmental regulations, and changing system capability requirements. Risk management continues to be used to identify, monitor, and mitigate all of the technology refresh-related risks.

Although during the earlier stages, technology refresh plans identified the expected cycles, planning must be revisited throughout the system lifecycle to periodically reassess the system to determine any new technology refresh requirements. These may be a result of new technologies and identify potential obsolescence that had not been identified in the earlier stages. The LCSP most

likely will require revision to reflect those impacts as well as changes in operational needs, evolving threats, process improvements, and plans for follow-on systems or any combination of these. All of these have the potential to affect technology refresh plans.

Trade studies that include market research, performed in the early stages, continue to be performed during these stages to continually identify potential products and services that may support or enhance the system performance and sustainment activities. Trade-off analysis should include the consideration of system performance, system availability (reliability, maintainability, supportability and producibility), process efficiency, and system lifecycle costs.

Appropriate planning for the insertion of the technology refresh is critical during these stages to balance the logistics workload, system needs, and risks, and ensure that disruptions to the operational mission are minimized and resources are optimally utilized. In addition, planning needs to consider as to whether additional verification or validation tasks will be needed to ensure the technology refresh performs to expectations. Should additional verification or validation be required, engineering may need to revise the existing test plan and procedures to support these efforts.

24.6 Technology Refresh in Ground Systems

Technology refresh in ground systems is a process, not an event. Driven by cost and technology factors, current and future ground systems are dependent on a large and growing number of commercial, interdependent hardware and software components, each with its own release and upgrade cycle, and end-of-life moments. The goal of technology refresh is multifaceted: to keep up with technology advances and avoid supportability issues without breaking anything or alienating stakeholders. There is no single solution to the problem, merely a set of considerations based on the architecture and the operational concept. A set of architectural principles that is assumed to underlie a future satellite ground system or other computer-communications platform should be considered

24.6.1 Architecture and Governance

Figure 24-2 is a representative stack architecture [17]. The various agencies involved in ownership of the layers of the stack complicates the trade-off among the key refresh considerations. The focus of this representation is on governance, and it illustrates the challenges in coordinating technology refresh processes, planning, and implementation among a multitude of stakeholders.

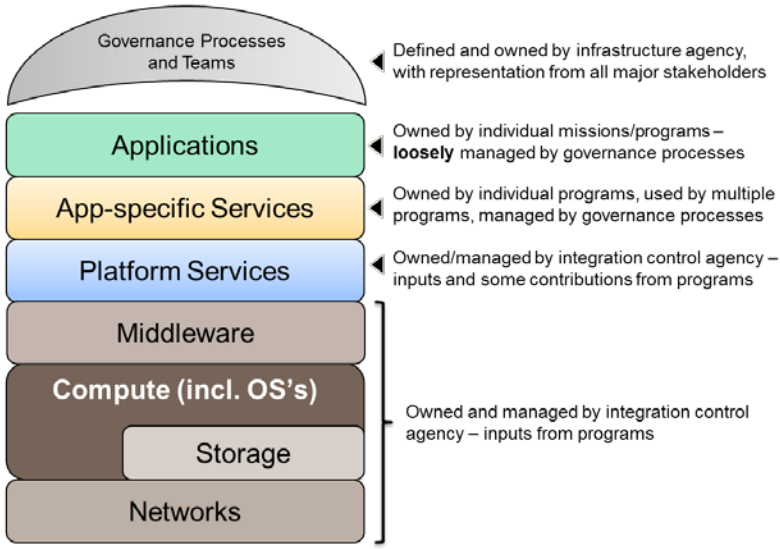


Figure 24-2. The stack: ownership and management.

24.6.2 Affordable and Resilient Ground Systems

The guiding principles for more affordable and resilient ground systems are illustrated in Figure 24-3. These were developed by means of a formal benchmarking of several government and commercial satellite operators, cyber resilience studies, and information technology trend assessments. Although the principles are intuitive, it is worth reading the report in its entirety [18]. The first four principles come from the benchmarking of commercial satellite operators (CSOs). The next three principles come from the findings of an associated commercial capabilities investigation. The cyber opacity and analytics and system resilience principles come from an associated cyber resilience and technologies investigation. The scalability and adaptability, agile management, and information accessibility principles come from an associated study of affordable development and deployment.

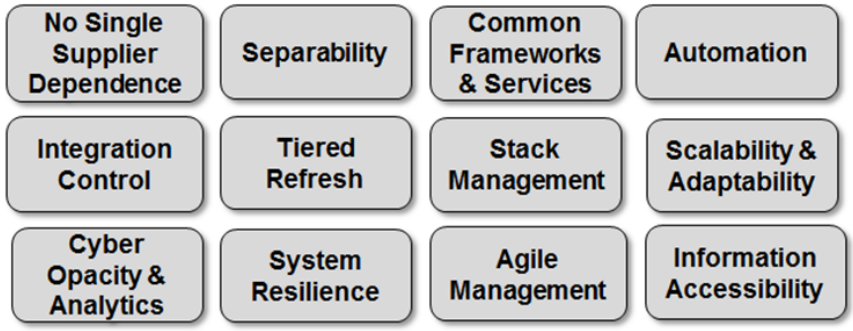


Figure 24-3. Guiding principles for more affordable and resilient ground systems.

24.6.3 Trade Space

There are four trades that the sustainment organization must make on behalf of, and in coordination with, the enterprise. There are endpoints of the trade space, and considerations that drive the sustainment organization to one portion of the trade space or another. The intent is that the trades involve the refresh of both hardware and software components, although not necessarily in the same time frame.

Trade #1: Stack evolution

The trade space is bounded by (1) an approach that upgrades the stack whenever one of its components is available for upgrade and (2) an approach that delays an upgrade until a problem occurs that a new release fixes or the current release is no longer supported. The second approach can be termed "If it isn't broken…." A version of the first approach that has been advanced is to stay one version behind to allow the upgrades to mature with broad use and have others find the problems. However, this does nothing about the frequency of upgrades but only introduces a hysteresis into the process. Consider the following factors when evaluating the options for this trade:

- Changing one component at a time follows the scientific method and thus facilitates the location of the root cause of any problem that surfaces.
- Waiting until a driving event occurs may not provide adequate time margin for making the upgrade, resulting in downtime that may impact the mission.
- Early installation of an upgrade or new hardware model should be considered if it has a new feature or capability that can benefit the

installation, such as labor cost savings, increasing needed margin, improving reliability or availability.

- Deferring an upgrade or new hardware model should be considered if it no longer supports a required capability. This will buy some time to consider whether the requirement can be abandoned or modified, or how it could be re-implemented by leveraging existing services that might not have been available or mature when the deprecated capability was initially provided.
- Consider whether a new feature comes at a cost, such as reduced performance, and assess the resultant trade. Cost may be secondary, for example, if the new feature improves cyber security, adopts a standard that is widely used by the enterprise and third party developers.
- Upgrades that affect the "look and feel" to the users and/or operators must be carefully considered as to the effect on training, documentation.
- Consider using complementary processes for upgrades, with incremental upgrades on a developmental system to isolate issues for resolution, and block upgrades of multiple components in the production system for stability and predictability. Provide access to users of the upgraded services on an increment by increment basis in a development environment so that they are able to test the compatibility incrementally and thus more readily isolate and diagnose incompatibilities. Nonetheless, production releases are regular, predictable, and non-disruptive.

⇒ ***Key Architecture Principles – Tiered Refresh, Stack Management, Integration Control***

Trade #2: Individual Stakeholders Autonomy Management

This trade is closely related to Trade #3, that of governance of the stack. Referring again to Figure 2, the more shared the technology, the less freedom individual programs should get

- The applications and the application-specific services layers are managed by the stakeholders of these applications using the governance processes of the enterprise.
- The platform services layer is managed by an integration control agency according to the governance processes of the enterprise. There are inputs from, and some services contributed by, the stakeholders.
- The middleware, computing, storage and network "layers" are also managed by an integration control agency according to the governance processes of the enterprise. There are inputs from the stakeholders.

- The integration control agency has representation from all of the stakeholders in developing and applying the governance process of the enterprise.

⇒ ***Key Architecture Principles – Common Frameworks & Services, Separability, Integration Control***

Trade #3: Governing the Stack

Governance drives the versioning and evolution of the stack. Governance is applied in three contexts: (1) the enterprise, (2) the infrastructure and integration control agency that operates and sustains the core infrastructure and shared functionality, and (3) the individual stakeholders (programs or projects) that utilize the enterprise services to meet their missions. The roles of each of these entities are generally captured below, and relate to the fact that local changes may have global impacts.

- Enterprise. Overall governance of the ability to accomplish the enterprise mission and its constituent elements, trade "outsourcing" of infrastructure to cloud providers vs. acquiring and operating it organically, establishing governance process. Develop overall services and infrastructure roadmap, migrating to common and shared services as necessary. Assess "new entrants" and GOTS.
- Infrastructure and integration control agency. Defines the initial set of application program interfaces (APIs), middleware, networks, data formats, and platform services standards, Assesses implementation options, schedule, cost trades for proposed new capabilities or requirements, follows project management template, considers whether a stakeholder change may benefit the enterprise if broadened and relocated[5]. Assess new technologies and data models. Assess "new entrants" and GOTS.
- Individual stakeholders. Convey needs to infrastructure and integration control agency, assess utility of new shared services and core services, implement mission-unique services and functions, concur with initial and evolving set of application program interfaces (APIs), middleware, networks, data formats, and platform services standards.

⇒ ***Key Architecture Principles: Integration Control, Stack Management and Agile Management***

[5] A proposed change in "program unique" applications still needs to be reviewed by the enterprise control agency. For example, the change may require a new service that could be provided to the entire enterprise rather than the one making the change, or conversely the change may provide a new service that could be provided to the entire enterprise.

Trade #4: IT Automating IT System Operations

There is not really much of a trade here. The system operators will wish, as much as possible, to control the provisioning and configuration of the key infrastructure resources (compute, storage, and networking) with software instead of hardware, and utilize automation as much as possible for that provisioning and configuration.

Table 24-1. Trades in IT System Operations Automation

Problem	Manual solution	Software assisted automation	Automation
Need additional tasks	Buy a new computer	Provision a virtual machine	Federation
Need additional storage	Buy a special-purpose disk array	Allocate from a storage pool	
Need new access patterns	Wire a bunch of computers together	Modify the network configuration in a browser	Automatic reconfiguration based on preset rules, including failover
Resource imbalance	Real-time monitoring	Alarms to operators using pre-set threshold	Automated priority-driven load rebalancing
Accommodate restricted data sources	"Stand up" an enclave with system high access	Reconfigure to isolate processing, storage, memory resources.	Defederation
Resource starvation, livelock, deadlock	Real-time monitoring	Alarms to operators or predefined scripts	Difficult to optimize

Moving to the right within this table lowers operations costs, and improves utilization, resiliency, and availability. It also impacts development, verification, and validation costs.

⇒ ***Key Architecture Principles – Automation, Stack Management***

24.7 Summary and Conclusions

Technology refresh is a process, not an event. As such, it must be planned and managed from the program initiation and throughout the product lifecycle to help mitigate risks that will significantly impact both hardware and software

component operations, maintenance, support, design and certifications. Ideally, planning for technology refresh needs to start during the concept development stage and should be defined for each successive program increment from the start of the system lifecycle. Furthermore, planning must be revisited throughout the system lifecycle to periodically reassess the system to determine any technology refresh requirements.

With the evolution to enterprise ground systems, planning for the technology refresh of future ground systems will be more regimented to address the increased number and diversity of stakeholders and an architecture that enables a service-oriented approach. Although this evolution will have a positive on lifecycle costs, its management needs to take into account a number of guiding principles that are noted in this chapter. These principles, together with the specific needs and circumstances of the hardware and software involved, will drive the four trade areas in one direction or the other. A "one size fits all" solution does not exist.

24.8 References

1. Federal Acquisition Register. Small Entity Compliance Guide. FAC 2005-73. May 29, 2014.

2. Department of Defense, DOD Supply Chain U.S. Materiel Management Procedures Operational Requirements, DOD-4140.01-R.

3. AFI 63-101/20-101. Integrated Life Cycle Management. March 7, 2013.

4. DODI 8330.01, Interoperability of Information Technology (IT), including National Security Systems (NSS). May 21, 2014.

5. Public Law 106-398. National Defense Authorization, Fiscal Year 2001. October 30, 2000. Washington, DC, Government Printing Office.

6. Office of Management and Budget, E-Authentication Guidance for Federal Agencies. OMB M-04-04. December 16, 2003.

7. DODD 5000.01, The Defense Acquisition System. November 20, 2007,

8. DODI 5000.02, Operation of the Defense Acquisition System. January 7, 2013.

9. NIST Special Publication (SP) 800-63-2, Electronic Authentication Guideline. August 2013.

10. U.S. Department of Commerce, National Institutes of Standards and Technolgoy. *Guide for Applying the Risk Management Framework to Federal Information Systems/A Security Life Cycle Approach*. NIST SP 800-37. Galthersberg, MD. 2001.

11. Tomczykowski, Walter. Defense MicroElectronics Activity (DMEA) DMSMS Acquisition Guidelines: Implementing Parts Obsolescence Management Contractual Requirements, Annapolis, MD: ARINC, Rev 3.0/ 2001.

12. IEEE Computer Society, IEEE Guide-Adoption of ISO/IEC TR24748 (Parts 1-4), Systems and Software Engineering — Life Cycle Management, New York, NY. The Institute of Electrical and Electronics Engineers.

13. Defense Acquisition University. Manager's Guide to Technology Transition in an Evolutionary Acquisition Environment. Fort Belvoir, VA; Department of Defense, Defense Acquisition University Press. June 2005.

14. Defense Acquisition Guidebook. Dag.dau.mil/Pages/Default.apx.

15. Unell, Alan D. et al., Evaluating Software Architectures in Space and Ground Systems. ATR-2012(9010)-12. The Aerospace Corporation, El Segundo, CA. July 17, 2012.

16. Abelson, Linda, et al. *Integrating Software Topics into the Request for Proposal*. TOR-2011(8506)-117, The Aerospace Corporation, El Segundo, CA. 2011. July 19, 2012.

17. Dashofy, Eric, Framework for an Affordable and Resilient Saellite Ground Enterprise for National Security Space Missions. Presentation at ASMC Focus Day. August 2014.

18. Campbell Asya et al., Applying Guiding Principles in the Development of Architectures, Acquisition Specifications, and Operating Practices for Affordable and Resilient Satellite Ground Systems. Proceedings of the Ground Systems Architecture Workshop 2015. March 3, 2015.

24.9 Acronyms

AFI	air force instruction
API	application program interfaces
ASMC	Aerospace Strategic Management Committee
CI	commercial item

COTS	commercial off the shelf (item)
CSO	commercial satellite operators
DAG	Defense Acquisition Guidebook
DMEA	defense microelectronics activity
DMSMS	diminishing manufacturing sources and material shortages
DOD	Department of Defense
DODD	Department of Defense Directive
DODI	Department of Defense Instruction
FAR	Federal Acquisition Regulations
FOSS	free and open source software
GOTS	government off the shelf (item)
IB	industrial base
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IMP	integrated master plan
IPS	integrated product support
ISO	International Standard Organization
IT	information technology
LCMP	life cycle management plan
LCSP	life cycle sustainment plan
MOSA	modular open systems approach
NDI	non-developmental item
NIST	National Institute of Standards and Technology
NSS	National Security Systems
OMB	Office of Management and Budget
OS	operating system (software)
R&D	research and development
RF	radio frequency
RFP	request for proposal
SDP	software development plan
SEMP	systems engineering management plan
SEP	systems engineering plan
SP	special publication
STP	software transition plan

Chapter 25
Risk Management

Andrew Y. Hsu
Acquisition Risk and Reliability Engineering Department
Mission Assurance Subdivision

25.1 Introduction/Background

Risk refers to the possibility of future events that, if they occur, would result in an adverse outcome. As such, risk is comprised of three components: (1) the event or chain of events that would lead to an adverse outcome, (2) the likelihood that the event or chain of events would occur, and (3) the impact or consequence of that outcome if realized.

Risk management (RM) is a process that supports program management functions. The RM process aims to identify, assess, and address risks to the execution of the program. The consequences of the risks are generally defined in terms of cost, schedule, or performance impacts. RM typically calls upon many other mission assurance disciplines throughout the process and can serve as a way to organize and communicate the findings of these other disciplines.

RM can be used in both a qualitative and a quantitative sense. By defining qualitative consequence and likelihood ratings, risks can be assessed, at least on a preliminary level. The most significant risks can then be filtered down for more rigorous quantitative assessments. Quantitative risk assessment methods can take on many forms; however, it should be noted that quantitative risk assessments do not necessarily imply a high level of precision. The value of applying quantitative techniques is in the objectivity that it brings. The level of uncertainty in quantitative estimates should ideally be communicated to decision-makers through the use of uncertainty ranges. The RM process allows users to prioritize risks, evaluate risk control plans, and track progress in reducing risk over time.

RM provides stakeholders with a reasoned, objective way to identify, assess, prioritize and control the uncertainties that a program faces. Prioritizing and preventing risks from occurring or mitigating their consequences early ensures avoidance of costly changes in the later stages of the program acquisition and operational lifecycle. DoDI 5000.02 requires program managers to identify top program risks and associated risk mitigation plans in the program acquisition strategy, and to present risk status at all relevant decision points and milestones [1]. A RM process identifies and communicates threats to mission success to decision makers and program stakeholders at all levels.

25.2 Definitions

Risk An expression of an uncertain event that could have a negative impact on a program. In general, risk consists of an event or sequence of events which are uncertain to occur, the likelihood of the event(s) occurring, and the consequence of the event(s) if realized. Risk is often expressed as an if-then statement in the form of *if [the uncertain event] occurs, then the program will experience [a specific set of consequences]*.

Likelihood The degree of belief that the uncertain events of a risk will occur, usually expressed in terms of a probability.

Consequence The negative effects on a program if the uncertain events do occur. These are usually expressed in terms of cost, schedule, or technical performance. Also called **impact**.

Risk prevention An approach to address risk by reducing the probability of occurrence of a risk's associated sequence of events.

Risk mitigation An approach to address risk by reducing the consequences of a risk given that the associated sequence of events has occurred.

Risk handling plan A plan to address a risk, typically by either preventing the associated sequence of events from occurring or mitigating their consequences to the program. Also called a risk mitigation plan (this is a misnomer because "mitigation plans" can also include preventive measures).

Risk scenario A system or mission condition that can be formally described as a cause-effect sequence of events, the occurrence of which may cause a mission risk impact and associated consequences to be realized.

Mission Impact The consequence to mission performance if a risk is realized. This is typically measured in objective terms by a performance parameter associated with a particular mission performance domain.

25.3 Objectives

RM provides a structured way to understand and address the negative effects of the uncertainties that a program faces. These negative effects can generally include technical performance shortfalls, increased cost, or schedule delays, or some combination of these. From a systems engineering and mission assurance standpoint, the negative effects of interest are technical performance in nature. Several lower-level objectives can then be defined:

- Systematic identification of concerns that have potential impact on successful mission execution
- Formulation and use of explicit criteria and means of evaluation to decide whether mission assurance actions are necessary regarding the identified risks
- Selection, execution, and tracking of risk controls that optimize the level of risk reduction given the resource constraints of the program.

25.4 Practices and Core Activities

Programs define a structured RM process in a formal risk management plan (RMP). This document describes the steps and cycles of the risk management process and assigns roles and responsibilities. ISO 17666 *Space Systems—Risk Management* and the *DOD Risk Management Guide for Acquisition Programs* provide guidance in the formulation of the RMP [2,3].

RM core activities include: risk identification; risk analysis, risk mitigation; and risk monitoring, as depicted in Figure 25-1. The RM process should be preceded by a planning activity where program-specific tailoring of the general risk management process occurs, and formally documented in the program RMP.

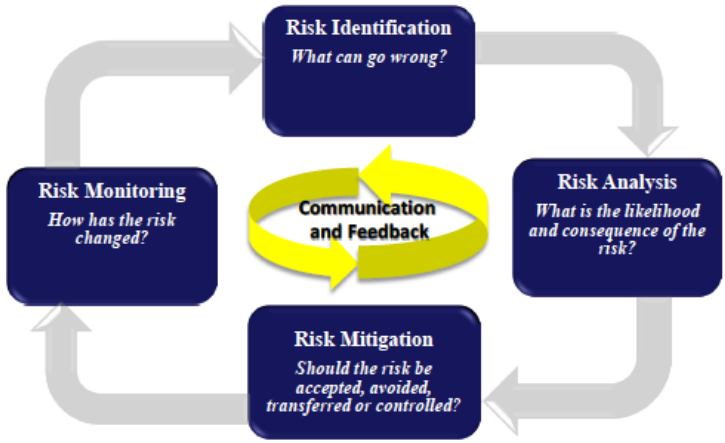


Figure 25-1. Risk management process (adapted from [3]).

25.4.1 Risk Planning

Risk planning includes defining a RM process to meet the specific needs of a program. The end product of risk planning is typically an RMP, which describes the program's risk activities. A good RMP should:

- Describe an organized, comprehensive, and integrated approach for managing risks
- Define the goals, objectives, and the program's risk management processes
- Describe an approach to identify, analyze, mitigate, and monitor risks across the program
- Define the methods and processes used to execute a program's RMP
- Help the program plan for adequate resources, including personnel, schedule, and budget [3]

Of particular importance is the definition of the likelihood and consequence scales and categories. Careful calibration of the consequence dimensions (cost, schedule, and performance) is needed so that one dimension is not more highly weighted at the expense of the others, particularly when different risks are collectively presented.

25.4.2 Risk Identification

A fundamental step in any risk management process is identifying the unknowns that could negatively impact a program. While risk identification often occurs in an ad hoc manner with technical subject matter experts raising concerns as they encounter them, a systematic search for risks is also possible through the use of tools such as master logic diagrams. Typical risk-identification methods are outlined in Table 25-1, grouping the methods based on personal experience and knowledge (experiential), and based on analysis of data (analytical). Each method has advantages and disadvantages. Effective risk identification usually requires a combination of two or more methods to overcome the disadvantages of any single method.

Table 25-1. Risk Identification Methods

Risk ID Method	Description	Strengths	Weaknesses
Experiential Methods			
Review previous program risks, issues, and lessons learned	Review of risks and issues identified on prior programs of similar scope, complexity, and use of technologies to see if any are applicable to the current program.	Leverages relevant knowledge from similar programs.	May not include risks outside of prior programs' experiences. Differences between programs may not be understood.
Checklists and questionnaires	Structured method to identify known potential risk areas based on past experience, and to have responders assess the applicability of those potential risks to the current program.	Leverages institutional and organizational lessons learned	May not identify risks outside of the group's prior experiences. Requires organization repository and maintenance.

Risk ID Method	Description	Strengths	Weaknesses
Brainstorming	Utilizes social interaction to enhance the risk identification process. It requires a competent and unbiased facilitator to help keep the discussion on topic.	Provides a structured method to leverage the knowledge breadth of a diverse group of experts	Dominating individuals may attempt to push their ideas onto the rest of the group, and weaker personalities might not get a chance to air their views. Only as good as the experience breadth of the group.
Personal knowledge/ experience of risks	Collect risks based on one or more individual's personal knowledge and expertise. Example questions "What are you worried about? What keeps you up at night?"	Beneficial within each individual's experience range	Limited team experience or knowledge. Individuals can inject biases into process. May not capture institutional experience lost with attrition. May not translate experience, design weakness, etc. into a risk framework.
Analytical Methods			
Risk ID Method	**Description**	**Strengths**	**Weaknesses**
Key Performance Parameters (KPPs)— technical, programmatic	Review of the KPPs to identify the specific risks to achieving the key program objectives. Monitor trends in KPPs and margins/reserves.	Provides risk identification that is targeted on the design's ability to meet the program's KPPs	Assumes the program's identified KPPs fully represent the parameters that best represent the required system performance.
Review Project Work Breakdown Structure (WBS)	A critical review of the WBS can expose risks inherent in the interdependency of the project work	Provides a structured approach for risk identification in the context of how the program's work is structured, including entities external to the program (suppliers, teammates, govern-mental entities, etc.)	Risk identification using the WBS is only as good as the WBS itself, and the expertise of the risk identifiers reviewing the WBS.
Risk Breakdown Structure	Risks are stated and assessed at each level of architectural assembly: system, subsystem, unit, component and part. Higher-level risk assessments are informed largely by historical data. Middle level risks also include the risk of interface and interaction. Component and part level risks are only assessed for very high, unit-level risks.	Comprehensive, structured, and intuitive for the reviewer. Aggregate risks include the probabilistic sum of all of the constituent elements.	Aggregation is subjective, and typically not statistical or mathematical—resulting in decreased confidence. Low aggregate risks may mask high concen-trations of risk in certain components or parts. Effective mitigation is sometimes best performed at a different level than the level being reviewed.

Risk ID Method	Description	Strengths	Weaknesses
Inception Risk Standardization	Each program assesses and dispositions a list of pre-defined standard risks based on the experience and data collected from historical programs and missions	This method requires programs to assess likely risks which may be overlooked.	Pre-defined standardized risk lists are not likely to be insightful to mission and program specific risks.
Review Requirements, Design Documents, and Drawings	Review of these documents can reveal perceived gaps in the design, or over-constraints that could adversely affect design development	Provides a structured approach for risk identification in the context of the program's requirements and design documentation.	Risk identification using the requirements and design documentation is only as good as the documentation itself, and the expertise of the risk identifiers reviewing the documentation.
Utilization of Models and Simulations	Early models and simulations can help identify weak points in the requirements or the design, and help direct programmatic attention to address concerns before they manifest as design issues	Models and simulations provide early insight into the design and its performance, from which risks (and issues) can be identified and documented.	Risk identification using models and simulations depends on how well they correlate to the actual design, level of realism, and the expertise of the risk identifiers analyzing and interpreting the results.
Fault Tree Analysis (FTA) and/or Root Cause Analysis (RCA)	FTA provides insight into design weaknesses and helps the engineering team identify added mitigations that may prevent faults or minimize impact of faults. RCA provides insight into process weaknesses and helps organizations add mitigations that may prevent fault recurrence.	FTA and RCA provide a rigorous methodology to understand potential contributors to a given fault. The process could help inform the analyst as to where a design is exposed to otherwise unidentified risks.	Risk identification during the FTA or RCA process depends on the depth and breadth of the analysis, and the expertise of the analyst. RCA responds to the presence of a failure and can be useful in predicting recurrence, but are not useful in predicting first occurrence.
Failure Modes and Effects Analysis (FMEA)	FMEAs help identify where design is exposed to failure modes, and inform the program on technical risks, consequences, and need for added mitigation	FMEA provides a rigorous methodology to identify and understand the failure modes of a given design. This better informs the program's risk identification process both at the unit/ subassembly level, as well as at the system level.	Risk identification during the FMEA process depends on both the rigor applied to the FMEA, and the systemic under-standing of how a unit's failure modes/effects will impact performance of the larger system.

Risk ID Method	Description	Strengths	Weaknesses
Review of Test Plans or Test Results	Test plan reviews (for breadth and depth of testing) help to identify where a system test plan may be inadequate in ensuring requirements are addressed and properly verified; Test results reviews help to identify if risk has been realized, and may also inform the engineer of unexpected performance attributes that pose potential risk to system performance.	Reviewing test plans in the context of risk identification can provide the reviewer insight into verification risks. Reviewing test data in the context of risk identification can provide the reviewer the first opportunity to assess any unexpected actual performance of the element under test, and evaluate its potential risk to the larger system.	Risk identification derived from test plan reviews tend to focus only on what is tested (as opposed to what is not tested). For test data reviews, a reviewer may unintentionally mask a discovered issue as a risk.
Assessing exceptions to mission assurance standards and processes	An evaluation of tailorings, waivers, or deviations from Customer or enterprise required mission assurance standards and processes to assess risk potentially introduced by these exceptions.	Establishes risks relative to an accepted baseline. Performing to modified standards may have inherent risks, unidentified.	None

Significant barriers to risk identification also exist due to a myriad of causes. Descriptions of these barriers and potential solutions are offered in more detail in TOR-2014-02201, *Technical Risk Identification at Program Inception* [8].

Potential risk areas that are of special interest to ground systems include: the maintainability and availability of the ground system, the potential for human error in commanding space assets, the processing and dissemination of data from space assets, and cybersecurity.

25.4.3 Risk Assessment

The risk assessment step in the RM process involves the evaluation of the likelihood and consequences of the identified risks. This can either be done according to qualitative or quantitative scales (see Figure 25-2 for an example set of risk scales). Qualitative measures such as a traditional qualitative 5×5 risk matrix can be used initially to screen risks for those that are more significant or of particular interest to program management. Making decisions under such circumstances is inherently subjective, but the quality of those decisions can be greatly improved by basing them on objective assessments. This necessitates the incorporation and implementation of a more rigorous, quantitative risk assessment process than the traditional qualitative risk matrix.

Bin / Category	Qualitative Descriptor	Quantitative Probability OOM *	Quantitative OOM Definition
A	Very Low	p ~ 1/1000 per mission	0 < p <= 3/1000 per mission
B	Low	p ~ 1/100 per mission	4/1000 < p <= 3/100 per mission
C	Medium	p ~ 1/10 per mission	4/100 < p < 2/10 per mission
D	High	p ~ 1/3 per mission	2/10 <= p <= 3/10 per mission
E	Very High	p ~ 2/3 per mission	4/10 < p <= 1 per mission

* OOM = Order of Magnitude

Likelihood of Occurence	Severity of Consequence 1	2	3	4	5
5	5	10	15	20	25
4	4	8	12	16	20
3	3	6	9	12	15
2	2	4	6	8	10
1	1	2	3	4	5

Bin / Category	Summary-Level Qualitative Descriptor	Quantitative Order of Magnitude of Outcome Severity	Qualitative Severity Definition / Assessment in Mission Performance Terms	Qualitative Severity Definition / Assessment in Mission Schedule Terms
1	Very Low	~3% Performance / Schedule Utility Loss (1-5% range)	Minor deviation from required performance of primary mission	Minor deviation from required primary mission schedule
2	Low	~10% Performance / Schedule Utility Loss (6-15% range)	Significant but non-impairing deviation from required performance of primary mission; or Impairing deviation from required performance of secondary mission	Significant but non-impairing deviation from required primary mission schedule; or Impairing deviation from required secondary mission schedule
3	Medium	~30% Performance / Schedule Utility Loss (16-45% range)	Impairing deviation from required performance of primary mission; or severely-impairing deviation from required performance of secondary mission	Impairing deviation from required primary mission schedule; or severely-impairing deviation from secondary mission schedule
4	High	~60% Performance / Schedule Utility Loss (46-75% range)	Severely-impairing deviation from required performance of primary mission; or failure of secondary mission	Severely-impairing deviation from primary mission schedule; or massive schedule delay makes secondary mission worthless
5	Very High	~100% Performance / Schedule Utility Loss (86-100% range)	Failure of primary mission	Schedule delay makes primary mission worthless

Figure 25-2. Example likelihood and consequence scales (adapted from [4]).

25.4.3.1 Cost Risk Analysis Methodology

Cost-risk analysis evaluates both the cost baseline and potential excursions, and includes cost-estimating uncertainty, technical risk (and potentially schedule risk) to statistically adjust the reference point estimate (most likely cost), and development of a cost vs. probability function. Monte Carlo simulation is typically performed to combine the individual lower-level probability distributions. Software packages such as @Risk™ and Crystal Ball™ are used to augment Microsoft Excel™ for these purposes.

25.4.3.2 Schedule Risk Analysis Methodology

The schedule risk assessment process evaluates both the baseline schedule and potential excursions, and incorporates schedule-estimating uncertainty and technical-risk information. Additional metrics are used to assist in evaluating the results, including the risk-driven critical path (at the desired confidence level) and the percent time on the critical path. This information is used to support

developing excursions to the integrated management schedule and cost as an independent variable (CAIV) trades and to assist in selecting risk handling strategies. Schedule quantitative risk analysis is performed during the program using Monte Carlo simulations derived from add-ins to Microsoft Project (e.g., @Risk for Project and Risk+).

25.4.3.3 Mission Risk Analysis Methodology

A quantitative mission risk assessment methodology depicts the potential outcome of a risk as a series of events, with each event having an associated likelihood, and each outcome having an associated mission impact [6]. Each event likelihood and outcome mission performance consequence can then be modeled with a probability distribution; each distribution encodes the level of knowledge available for each parameter. This feature is significant in that the uncertainties affecting the assessment of the risk are now made visible.

25.4.3.3.1 Risk Scenario Development

The goal for the development of a risk scenario is to diagram the sequence of events that must occur in order for each risk to be realized (Figure 25-3). This forces risk practitioners and subject matter experts to explicitly define the logic of the failure mechanism. Pivotal events or conditions that affect the outcome must be carefully considered because often a risk is realized not as the result of a single event, but rather as a result of an initial event occurring in conjunction with a number of intermediate events. In building the scenario event sequence some of the intermediate events could actually be identified as targets for preventive or mitigative risk control measures, thereby reducing the likelihood of the risk occurring.

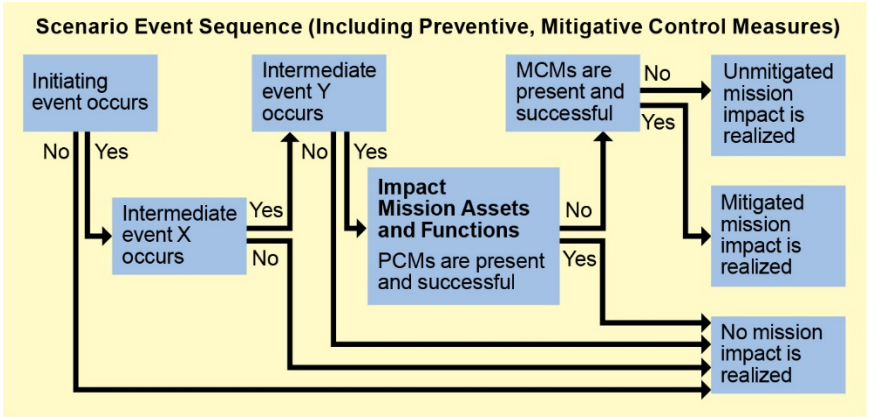


Figure 25-3. Risk scenario [6].

These risk scenarios provide a common understanding of the sequence of events and how the events affect the outcome of a risk as an impact to mission assets and functions. This in turn allows for a more structured discussion of how to model the risk by decomposing it into its constituent events. Developing the risk scenario can also prevent an overly singular focus on the initiating event (root cause) in developing control measures. Because stopping any of the events in the chain would prevent the risk, a risk control measure can then target the most preventable event or events in the scenario.

25.4.3.3.2 Quantitative Assessment

Once the risk scenario sequence of events is defined, the probability of the initiating event, conditional probabilities of the intermediate events, and the severity of the mission performance impact for each outcome can be quantitatively assessed. Given that the risk has *n* discrete outcomes, the risk can then be expressed as a set of *n* probability-consequence pairs. The probability of an outcome *i* (P_i) is expressed in terms of the likelihood of the initiating event *P_init* and the conditional likelihoods of the intermediate events that lead to outcome *i*:

$$P_i = P_{init} \times P(A|init) \times P(B|init, A) \times P(C|init, A, B) \ldots \quad (25.1)$$

Ground systems are typically also concerned with availability rates (the fraction of time that a system is operational over some period of time). In this case, availability can be substituted for probability in the above equation.

The mission impact of an outcome is expressed in terms of the shortfall in mission performance through the life of the mission. Performance shortfalls are normalized to a scale on which 1 corresponds to complete mission loss and 0 corresponds to no mission impact. The mission impact is normalized over the entire mission life. So, for example, if a risk results in the loss of half the utility of the payload over the last year of a ten-year mission, the mission impact is 0.5 * (1 year/10 years) = 0.05.

The mission impact of a deferred or delayed ground system capability can be assessed using the same approach. For example, a ground system capability that enables the use of an on-orbit payload that was originally planned to be available at launch is delayed by a year. The payload provides 25% of the utility of mission. The mission impact is then 0.25 * (1 year/10 years) = .025.

The risk practitioner can facilitate these assessments, but must rely on domain experts and program management to provide the necessary information to perform the evaluation. This information can take the form of test or field data, other analyses, or engineering judgment. Uncertainty in each probability and

mission impact parameter can be modeled by defining the parameters as probability distributions.

This process further provides the opportunity to evaluate and communicate technically significant mission-impacting risks. Risks essentially can be screened to identify those that are potentially mission impacting. Additional assessment and quantification techniques can then be applied to those impacting major technical risks.

Qualitative risk management processes are often based on subjective definitions. A quantitative risk assessment approach with objective measures allows decision-makers to consider, compare, and prioritize objective assessments of risk and improve decision quality. Thus, risk quantification is utilized for objectivity, not for precision. A detailed bottom-up quantification of reference scenarios methods requires iterative brainstorming to capture event sequences and relevant parameters.

25.4.3.3.3 Risk Communication

Each risk can be plotted on a probability vs. mission impact risk map. Figure 25-4 shows a standard risk map used for a typical Class A (very risk-averse) mission. Standardized risk communication is a useful decision support aid to provide to program managers. The risk averseness of the program should be reflected in the likelihood and impact scales. For example, because many risks in the aerospace industry are low-likelihood but high-consequence, the likelihood scale emphasizes very small probabilities.

If special requirements exist for a mission segment or component, the risk reference boundaries for evaluation of the related risks need to be defined and set up to reflect those boundaries in the impact risk map. For example, the breach of a ground-system availability requirement (typically 99%) may be unacceptable in terms of the expected quality of that ground system, but would normally translate to a relatively minor impact on overall mission performance. In this case the risk would need to be evaluated in terms of not satisfying the applicable rules or requirements and the risk matrix may not make sense as a way to communicate the breach of a requirement. Special assessments may be needed and should be performed in objective, quantitative terms. Programs should strive for clear, unequivocal description of risks.

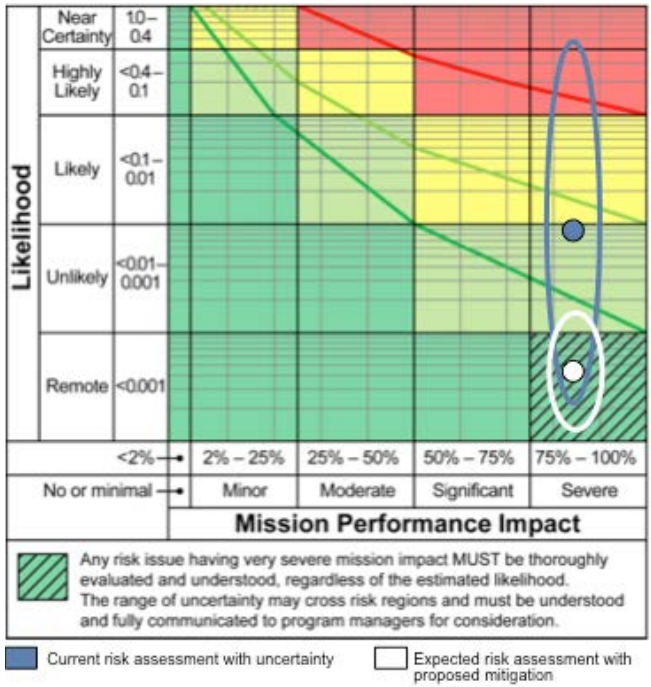


Figure 25-4. Quantitative risk map [8].

25.4.3.4 Risk Handling

Once risks are understood and assessed, a program can then apply measures to reduce either the likelihood or consequence (or both) of the significant risks. These measures are described in risk handling plans. Candidate risk handling plans are typically evaluated in terms of feasibility, expected effectiveness, cost and schedule implications, and their effects on the system's technical performance. A plan should detail the assumptions made in the plan, the steps that must be completed and associated schedule, the resources required, the cost of implementing the plan, and recommended metrics and key milestones for tracking progress.

Risk handling plans generally follow four strategies:

- Avoidance
- Control
- Transfer
- Acceptance

Avoidance typically means that changes to the CONOPS, requirements, specifications, or processes are made to circumvent the risk entirely. Risk *controls* work to reduce the likelihood of a risk (prevention), the consequence of a risk (mitigation), or a combination of the two. *Transfer* involves the reallocation or redistribution of requirements or other causes of a risk to another part of the system. For example, the risk of implementation of a required function within software could be transferred to hardware and vice versa. *Acceptance* is the explicit acknowledgement of the existence of a risk and assumption of any consequence that may be incurred if the risk is realized. This typically is used for low risks, with a caveat that they be monitored in case the situation changes such that the risk likelihood increases.

A program's decisions in choosing a portfolio of risk handling plans should optimize the level of risk reduction subject to budgetary and other resource constraints that the program faces.

25.4.3.5 Risk Monitoring

Risk monitoring involves the tracking of progress in risk reduction as risk handling plans are implemented and executed over time. If the progress deviates significantly from planned, then a revisit of the risk handling options could be in order. Risk monitoring is accomplished through three primary tasks:

- Systematic and regular tracking and evaluation of the execution of risk handling plans to determine if the plans are having the expected effect in reducing risk.
- Repeating the risk management cycle at regular intervals or as needed so that adjustments to current risks can be made and new risks can be identified.
- Re-optimization of the risk handling portfolio based on the completion or adjustment of existing risk handling plans and the introduction of new risks.

Continuous risk monitoring assures that the risk management process keeps abreast of program development and that the overall set of risks, assessments, and handling plans evolves as the program evolves.

A relatively simple way of monitoring the risk at the program level is by tracking the aggregate risk exposure of the program:

$$RE = \sum_{i=1}^{n} P_i \, C_i$$

where n is the total number of risks tracked by the program, P_i is the probability of the ith risk occurring, and C_i is the impact of the risk. By tracking this

measure over time, a program can show its progress in reducing its overall risk over time.

25.5 Key Lessons Learned

Some of the prime lessons learned in risk management (taken from the *Space Vehicle Engineering Handbook* [6], but are generally applicable to ground systems):

- The primary goals of program risk management are:
  - To support the development of an acquisition strategy to meet the user's need with the best balance of cost, schedule, and performance, and
  - To reduce the likelihood of not meeting program requirements by identifying risk events and dealing with them explicitly.
- Poor program planning will exacerbate a program's risk management efforts by establishing unrealistic objectives that do not adequately account for program risk.
- Risk identification is the most critical step in risk management, yet is often poorly done. Several systemic barriers prevent comprehensive risk identification. Typically, insufficient tools, techniques, and staff resources are available to comprehensively identify risks.
- Risk can only be assessed within the context of an acquisition strategy. Changing the acquisition strategy changes the risk.
- Unless the original plan was suboptimal, risk reduction will almost always involve trading between cost, schedule, and performance.
- Risk is defined in terms of cost, schedule, and/or performance dimensions. Under the CAIV concept, as cost tradeoffs (including risk) are made on an iterative basis, aggressive cost goals are established that become more of a constraint and less of a variable. Therefore, the PM may be required to trade performance and schedule (potentially increasing their associated risks) to meet CAIV cost constraints and reduce cost risk.
- Often risks are managed by lists that are ranked by subjective, qualitative measure. Quantification is crucial to making efficient and effective decisions in allocating resources to reduce risk. The resources required to implement a risk control should not outweigh the probable consequence of the risk. Qualitative measures can be used to screen risks for the more significant items. These should then be quantified to better inform risk-control decisions.
- Modeling and simulation is a key RM tool that is prospective in nature and useful for analyzing potential problems. However, it can also be a source of risk because predicted solutions to not always accurately match real-world performance.

- Risk can never be fully eliminated or completely transferred. A program's risk level cannot be reduced to zero. Therefore, risk must be prioritized for handling based upon reasoned assessment.
- The principal purpose of research and development is to reduce the uncertainty (and thereby the risk) associated with acquiring a new system. Some managers consider risk "good" in that acceptance of some risk allows opportunities for technological breakthroughs.
- Products developed throughout the risk management process must be captured as documentation for monitoring RM process activities.

25.6 Risk Management References

1. UnderSecretary of Defense for Acquisition, Technology, and Logistics Operation of the Defense Acquisition System. DoD 5000.2, January 7, 2015.

2. ISO 17666 Space Systems – Risk Management

3. Department of Defense Risk Management Guide for Defense Acquisition Programs, June 2015. (http://www.dau.mil/publications/publicationsDocs/RMG%206Ed%20Aug06.pdf)

4. SMC-G-1205 SMC Risk Management Process Guidance

5. NASA/SP-2011-3422 NASA Risk Management Handbook, November 2011.

6. Englehart, W. C., *Space Vehicle Systems Engineering Handbook*. The Aerospace Corporation, El Segundo, CA. 2005.

7. NASA/SP-2010-576. *NASA Risk-Informed Decision Making Handbook*, April 2010.

8. Hsu, Andrew Y., *Technical Risk Identification at Program Inception*. TOR-2014-02201, The Aerospace Corporation, El Segundo, CA. April 2014.

25.7 Bibliography

Guarro, Sergio B., *Mission Risk Assessment Process and Techniques for APR*. ATR-2012(9012)-1, The Aerospace Corporation, El Segundo, CA. October 2011.

TI-RA-1.1 Mission Risk Assessment Instruction for APR and ASMR Space Vehicle Engineering Handbook.

25.8 Acronyms

CAIV	cost as an independent variable
CONOPS	concept of operations
FMEA	failure modes and effects analysis
FTA	fault tree analysis
KPP	key performance parameters
RCA	root cause analysis
RM	risk management
RMP	risk management plan
WBS	work breakdown structure

Chapter 26
Configuration and Data Management

Leia R. Bowers
Software Acquisition and Modeling Department
Software Engineering Subdivision

26.1 Introduction/Background

This chapter discusses the processes and plans used to document and control the program baselines, including hardware (HW), software (SW), data, interfaces, procedures, and processes. Configuration management (CM) and data management (DM) processes provide the customer program office, contractor program managers, developers, and other team members the visibility they need into a program's configuration and data activities. The activities performed for CM and DM are not unique to the development lifecycle model being used or characteristic of the system.

CM is set of management controls within the context of the engineering process, rather than a set of specific activities performed or a set of organization functions. The customer and contractor team must have CM systems that are complementary and support the flow of information in both directions. CM tends to be more deeply involved in the software engineering process, and while the same general CM functions are performed for hardware, these controls are extended to include the process of developing a baseline.

CM is a fundamental requirement in all contracts to establish and maintain the integrity of work products using processes for planning and management, configuration identification, change management, status accounting, verification, and audit while supporting multiple information needs and responsibilities, including:

- Contractor program managers view CM as an aid in obtaining and supporting products consistent with the customer's evolving needs with a reasonable degree of confidence that the product is in the process of acquiring required attributes as the hardware and software are being developed and maintained. The program manager is responsible for authorizing resources to implement program objectives, controlling cost, schedule, and technical objectives, and coordinating team activities.
- Developers use CM processes to: establish baseline(s) to plan and measure product progress, support communication of interfaces (both external and internal), requirements and the integration of solutions to the customer's requirements. Developers are responsible for design integrity. Configuration item (CI) owners know and understand the design, provide

advice to others working on or interfacing with the CI. They serve as the technical control point for all CI modifications, including enhancement and repair. They ensure CI integrity by reviewing all changes and ensure periodic regression tests are conducted.

- The customer program office views CM as enhancing the likelihood that the product will perform as expected. CM precludes haphazard control of product development and maintenance, because the customer and developers may be required to show that they acted in a reasonable and prudent manner to ensure that adequate controls were applied (e.g., launch control software).
- The contractor's management use CM to understand productivity and the products as assets to be kept, maintained, and reused.

Government rights to data are specified in the contract. DM is initiated when a data item is created, revised, or when performing any of the actions necessary to transition it to the next status level and triggered to facilitate the flow of work throughout the lifecycle. The data is accessible in accordance with the contract and Contract Data Requirements List (CDRL) using application software available to the authorized users.

26.2 Definitions

Allocated baseline (ABL) Approved requirements for a product, subsystem or component, describing the functional, performance, interoperability, and interface requirements, that are allocated from higher-level requirements, and the verifications required to demonstrate achievement of those requirements, as established at a specific point in time and documented in the allocated configuration documentation

Approval authority Organization or person authorized to approve: a configuration change to a product; changes to product definition information and other related documents release (or cancellation) of documents for use anywhere or in a specific program; and commitment of resources

Change control Component of configuration management consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification

Configuration Product attributes of an existing or planned product, or a combination of products; one of a series of sequentially created variations of a product

Configuration audit Review of processes, product definition information, documented verification of compliance with requirements, and an inspection of

products, to confirm that products have achieved their required attributes and conform to release product configuration definition information

Configuration baseline Configuration of a product, at a specific point in time, which serves as a basis for defining change, for conducting verifications, and for other management activities. For a software product, the build baseline includes the actual product (e.g., functional baseline, allocated baseline, developmental baseline, build baseline, product baseline)

Configuration change Alteration to a product and/or its product configuration information

Configuration change management Function that ensures changes to and variances from a configuration baseline are properly identified, recorded, evaluated, approved or disapproved, and incorporated and verified as appropriate

Configuration control board (CCB) Group of people responsible for evaluating and approving or disapproving proposed changes to configuration items and for ensuring implementation of approved changes

Configuration identification CM function which establishes a structure for products and product configuration information; selects, defines, documents, and baselines product attributes; assigns unique identifiers to each product and product configuration information

Configuration item (CI) Aggregation of hardware, software, firmware or any discrete portion thereof that satisfies an end use function, and is designated for separate configuration management (i.e., it has specified requirements, and is an item to which the effectivity of changes is addressed)

Configuration management (CM) Technical and management process for establishing and maintaining consistency of a product's functional and physical attributes with its requirements, design and operational information throughout its life

Configuration status accounting CM function that formalizes the recording and reporting of the established product configuration information, the status of requested changes, and the implementation of approved changes including changes occurring to product units during operation and maintenance

Configuration verification Verification of requirements and incorporation of approved configuration changes

Contract Data Requirements List (CDRL) List of authorized data requirements for a specific procurement that forms a part of the contract. CDRLs provide a standardized method of clearly and unambiguously delineating the government's minimum essential data needs.

Data Recorded information that can include: technical data, computer software documents, financial and management information, representation of facts, numbers, or datum of any nature that can be communicated, stored, and processed

Data management (DM) Processes and systems that plan for, acquire, and provide stewardship for business and technical data, consistent with data requirements, throughout the data lifecycle.

Design release configuration Set of design information incrementally released to date by the developing activity for a product during a product's definition (development) phase

Document Self-contained body of information or data that can be packaged for delivery on a single medium (paper, photograph, digital files, optical computer disk, magnetic, electronic storage, or a combination thereof.) Examples of documents include: records, specifications, engineering drawings, drawing lists, pamphlets, reports, and standards.

Effectivity Designation defining the product range (e.g., serial numbers, block numbers, batch numbers, lot numbers, model, dates or event) at which a specific product configuration applies, a change is to be or has been affected, or to which a variance applies

Element Smallest product unit identified for individual development.

Enterprise Business or government organization or a discrete portion thereof. company, contractor, design authority, design activity, manufacturer, organization, supplier

Enterprise identifier String of characters that uniquely identifies an enterprise. This identifier is assigned by an issuing organization within a scheme and is unique to that scheme (e.g., commercial and government entity (CAGE) Code, DUNS, International Standards Organization (ISO) Supplier Code, and Organization Identifier).

Fit Ability of a product to interface or interconnect with, or become an integral part of, another product

Form Shape, size, dimensions, and other physically measurable parameters that characterize a product

Function Action or actions that a product is designed to perform.

Functional attributes Measurable performance characteristics expressed in terms of quantitative parameters (e.g., range, speed, lethality, reliability, maintainability, safety, processing speed, random access memory, operating and logistical parameters, including their respective tolerances where applicable, attributes, characteristics, requirements).

Functional baseline Describes the system's performance (functional, interoperability, and interface characteristics) and the verification required to demonstrate the achievement of those specified characteristics. It is directly traceable to the operational requirements contained in the Initial Capabilities Document (ICD) or equivalent document.

Functional configuration audit (FCA) Audit that formally examines each configuration item to ensure it meets the functional characteristics stated in its item performance specification. For a system as a whole, the audit examines the system functional characteristics has achieved the requirements in the allocated baseline.

Integrated data environment Tools for implementing digital data operations, may also be referred to as "Integrated Digital Environment".

Lifecycle Period of time that starts when a product is conceived and ends when the product is no longer available for use

Product Result of a process, including: hardware, software, processed materials, documentation, services, facilities and their combination into systems

Product baseline (PBL) Detailed design at a specific point in time, for production, fielding/deployment, and operations and support. The PBL prescribes all necessary physical (form, fit, or function) characteristics and selected functional characteristics designated for production acceptance testing and production test requirements. The initial PBL includes "build-to" specifications for hardware (product, process, material specifications, engineering drawings, 3D Computer Aided Design (CAD) models, and other related data) and software (software module design - "code-to" specifications). The As-Delivered and subsequent PBLs add product operational information needed to operate and maintain the product.

Product configuration audit (PCA) Audit that formally examinations the "as-built" configuration of a configuration item against its technical documentation to establish or verify the configuration item's product baseline

Tailoring Process by which decisions are made to exclude or modify individual requirements (sections, paragraphs, or sentences) in a standard or contract.

Technical data Product configuration information recorded (regardless of the form or method of recording) of a scientific or technical nature (including software configuration documentation) relating to supplies procured by an agency

Revision Attribute that distinguishes a change to a design or document in order to differentiate one closely related design or document iteration from another.

Rework Modification to a nonconformance that will completely eliminate it and result in a product that conforms completely to the drawings, specifications, or contract requirements

26.3 Objectives

The key objectives for discussing CM/DM include the following:

- Planning and implementing basic processes for CM/DM functions, including tailoring processes to select the items (e.g., documents, hardware, software, and data) to be controlled
- Initiating a baseline process
- Implementing effective and responsive change management
- Implementing data integrity and utility

26.4 Practices

26.4.1 Core Activities

Planning and management of work products and data is required to ensure the appropriate level for the configuration and data management activities throughout the product lifecycle. The planning activities include: the tailoring the processes and procedures to incorporate requirements from the contract and address how required stakeholder interactions will occur, document and get commitment to the plan, and maintain the plan throughout the lifecycle.

Identification is the basis from which the configuration of products are defined and verified, products and their product configuration information are labeled,

changes are managed, and traceability is maintained throughout the product's lifecycle. These processes begin as the system requirements are established.

Baseline initiation occurs when a document or product component has been formally reviewed and agreed to by responsible management.

- Internal baselines serve as the departure point for further development and can be changed only using change control procedures.
- External baselines with joint reviews by the developer/contractor and customer are created to minimize the discrepancy between what the customer wants/needs/expects and the delivered products.

Change control is the systematic evaluation, coordination, approval or disapproval, and implementation of approved changes to CIs after the configuration identification has been established and the document or product component is baselined. It ensures the integrity of baselines by ensuring the impact of change is understood and supports the escalation and resolution of issues prior to implementation.

Status accounting is the recording and reporting of information needed to manage configurations effectively. Status accounting includes: product tracking of approved configuration documents and identification numbers, tracking the status of proposed changes, deviations, and waivers to the review boards, disposition and implementation status of approved changes, and reconciling all CIs in the configuration of baselines. This data is used to generate status reports, analysis project processes and procedures, estimate future development or rework, and evaluate performance post deployment.

Verification and audit ensures the configuration of the product meets its requirements and matches its documentation.

Data management supports multiple stakeholders, including: space program operations (SPO), contractor program managers (PMs), developers and their management throughout the lifecycle. The types of data to be managed can be generalized to include: management, engineering, test and evaluation, operation, and project specific data, all of which are driven by contract requirements, product complexity, internal requirements and good engineering practice.

26.4.2 Command Media/Best Practices

CM and DM have long-term histories in most contractor companies that develop products for the Government. Industry standards that have guided industry's implementation include the following:

- EIA-649-1 Configuration Management Requirements For Defense Contracts, Nov 2014
- ISO 10007:2003 Quality management systems - Guidelines for configuration management
- Government Electronics and Information Technology Association (GEIA) Standard 836-2002 Configuration Management Data Exchange and Interoperability
- CMII-100E CMII Standard for Enterprise Configuration Management
- Capability Maturity Model® Integration for Development (CMMI®-DEV), Version 1.3, Configuration Management Process Area

Contractors have command media that include detailed organizational processes leveraging best practices that include: entry criteria, inputs needed, activities to be performed, outputs generated and exit criteria.

26.5 Key Lessons Learned

The following are several key points to keep in mind throughout the project lifecycle with regard to configuration and data management:

- Product development is inherently an evolutionary process. While change is a healthy and necessary part of product development, quality can be ensured only if change is controlled and documented in the development process, as well as the production process.
- Keeping track of a small product is no big deal; however, keeping track of large products requires planning as products get more complex and more critical.
- Ensure periodic updates to plans (i.e., CDRLs) to integrate improvements and address required lifecycle modifications.
- The development processes generate a sequence of documentation that describe CIs at different layers of detail and from different perspectives to identify all important attributes of the system. The documents record the functional and physical attributes of each CI in a controllable form for every CI produced. Ensuring identification relationships reduces time spent looking for documentation, minimizes ambiguity in the product, maximizes the product's internal consistency, simplifies defect correction and maintenance, and is required to build, release, and distribute products.
- Controlled change reduces confusion, duplication of effort, and rework. Without it, as ad hoc changes are continually made, a system becomes more resistant to change control because the impact is not understood, until each implementation of a change results in the need to make another change.
- Status accounting is easy if identification and control of changes is complete and accurate, including a description of what it is, where it fits

in the product architecture, time of creation, status of pending changes, revision history, and outstanding change requests with status (approved, deferred, rejected).

26.6 Task Execution by Phase

26.6.1 Principles

CM and DM are integral components of the product management and support. The processes to be used are influenced by the lifecycle management model selected for a project.

CM activities provide the focal point for system change ensuring authorized changes are recorded with sufficient detail to be reproduced or removed, and supports exception resolution.

DM processes are designed to archive and easily retrieve all project data not managed as part of the program baselines and gives project personnel the ability to progress as a team from concept to delivery. It supports the user's ability to apply the product in their work. It provides those that maintain the product with an ability to rebuild the product from its parts.

26.6.2 Execution by Acquisition Phase

The customer program office begins planning and specifying requirements for the management of work products and data at program initiation and the contractor's CMP is updated using data collected from performance of the program's CM/DM activities.

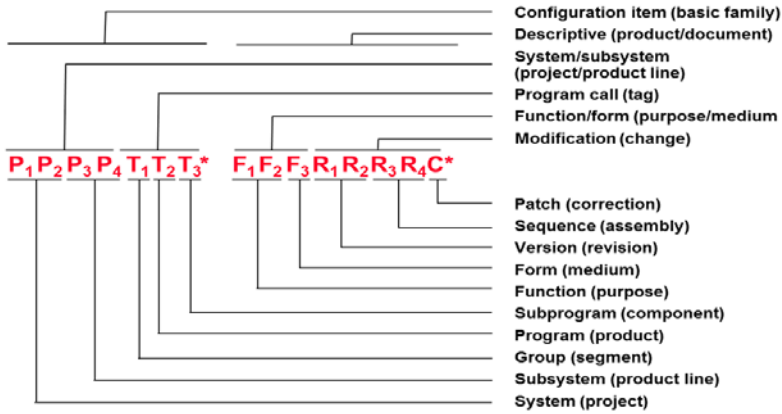
The contractor's planning of CM is required to ensure the appropriate level for the CM activities is applied throughout the lifecycle to meet contract and organization requirements. The contractor's configuration management planning may be described at a high level in the systems engineering management plan (SEMP) with sufficient detail for consistent performance described in the software development plan (SDP) and/or the hardware development plans (HDP). This includes selecting the work products to be controlled and the level of control to be applied, performance objectives (e.g., quality, time scale, cycle time, and resource usage), standards, contract requirements and schedule dependencies within the lifecycle activities, identifying resources (including funding, personnel, and tools), periodic evaluation and reporting requirements, establishing rules for identifying, baselining, controlling, and reporting of products and associated documentation. The resulting plans may be included in multiple documents (e.g., program management plan, configuration management plan, SEMP, hardware configuration management plan, software configuration management plan, and SDP); however, the approach, processes,

procedures, and measurements must be consistent across the entire team, including associate contractors, subcontractors, and suppliers.

Identification activities include: applying naming/numbering conventions to label all products/documentation, defining and updating the set of CIs, identifying and recording all dependencies, listing CIs in a baseline, cataloging all hardware and software engineering support tools.

A program identification number is assigned when the system requirements are identified and are used to derive the configuration numbering system to ensure all discrete units are assigned unique identifiers. The naming and numbering of component/document, models, prototypes, versions are used to associate the following information:

- Attributes – what it does, how it does it, where it does it, how fast it does it, what it needs to do it, what it produces, how big it is.
- Relationships – identifying consistency between representations requires all proposed changes must identify the impact and extent of the change. Knowing the relationships of information allows: planning when CIs should be controlled, minimizes the number of CIs handled during changes (i.e., less confusion, few discrepancies), reduces time spent looking for documentation, specifies actual CIs to be distributed, delivered, retained and assures propagation of approved changes during build, release, and distribution.
- A numbering scheme is used to identify all system components, layers of the system (e.g., requirements, design, and implementation), the CIs in each layer, the products, and all associated documentation/media.
    - Non-significant numbering systems are used when characteristics of the name are not expected to convey and information about the CI other than a unique identifier (e.g., 000001 – 999999 (sequential), 1501050900 (date-time of capture event). Using non-significant makes the process of assigning an identifier simple, and the same scheme can be used by multiple programs; however, records must be consulted to get any information about the item since the numbering does not provide any information.
    - Significant numbering systems use characters assigned to describe selected attributes of the product and the meaning of the character is position dependent, see Figure 26-1.



Key:
P, T, F and C alphanumeric characters
R numeric characters (0 - 9)
Character positions* may not be used simultaneously
No symbols or punctuation (except -)
No spaces between characters
Mnemonics used for external identifiers shall contain 8–14 characters

Note: MIL-STD-100G Standard Practice for Engineering Drawings includes the following restrictions: identifiers are no longer than 15 characters, preclude the use of I, O, Q, S, X, and Z, fractional, decimal or Roman numerals, blanks, or symbols.

Figure 26-1. Significant numbering schema.

- Version identification—information recorded to identify compatible and incompatible changes throughout a CI lifecycle, both for communication and enforcement purposes. Alphabetic variant with numerical successor (e.g., A1 → A2 → A3 → B1) or decimal numeric (e.g., 1 → 2 → 2.1 → 2.2 → 3) records are used to identify the origin of the variant.
  - Firmware identification – each programmable, read-only memory (PROM) is assigned a unique number and references the software/procurement numbers (e.g., requirements document, software specification, code, procured component specification), see Figure 26-2.

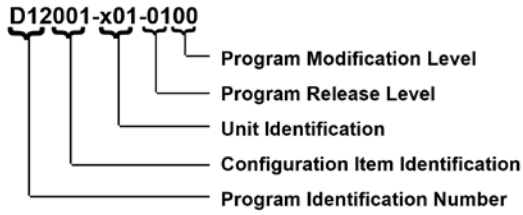


Figure 26-2. Firmware numbering schema.

- Documentation needed for a hardware PROM containing data that is not executable by a computer program include the part number of the uncoded device and a truth table.
- Documentation needed for a software PROM that is executable by a computer program include the part number of the unprogrammed device; the truth table of the coded device; and references to the design, source, and object code.
- Identification is the key to retrieval of CIs, documentation, and the relationships between CIs to aide in defect correction, correlating CI reuse, and lifecycle maintenance.

Baselines represent well-defined points in the product lifecycle activities where each CI is identified for development, created, verified and validated, accepted, cataloged and available for use or to interface with other CIs. Figure 26-3 illustrates the evolution each CI through a succession of lifecycle states.

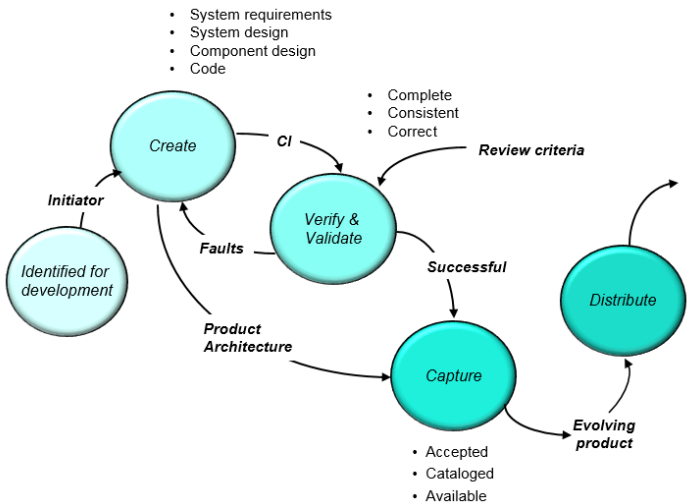


Figure 26-3. CI lifecycle states.

    - Internal baselines are configuration points established to define the evolving configuration of a product during the lifecycle. Internal baselines are frequently used during software development to describe the CI at any stage of the lifecycle, and to ensure associated documentation is under internal configuration control.
- The initial internal baseline is informally controlled (e.g., by developer or integrated product team [IPT]). Subsequent internal baselines are created as CIs and associated supporting data have been developed, verified, and validated. If the CI interfaces with other CIs it frequently is transferred from the informal control to a program resource library.
- Final internal baselines are created as CIs and associated documentation transition through the lifecycle are after control board review and approval for release to a static library.
    - External baselines are created for CIs as each phase of lifecycle produces more detailed representations of the system to improve the likelihood the customer will receive a useful product.
- Functional baselines define all the system level functions and system test criteria.
- Allocated baselines identify all requirements including design constraints and user-required standards.
- Product baselines define the exact version of the product including associated documentation to be accepted. Each new version will be identified with a version release number, a description of changes, an approved release document, identification of changes to any support equipment, and associated documents.

Change control identifies issues to be resolved before implementing the actual modification in the baselined product. The extent and formality depends on the nature of the change, the current lifecycle activities and the dependencies. Evaluating the impact of a change provides visibility into the required rework. A needed change may be the result of an identified defect, learning curve, a new requirement or emerging requirements as each lifecycle phase produces more detailed representations of the system. The impact may be isolated to a single CI or include multiple CIs across internal/external interfaces. The change may propagate changes through multiple lifecycle phases and activities that will result in release of a new product version. Figure 26-4 illustrates the change control process.

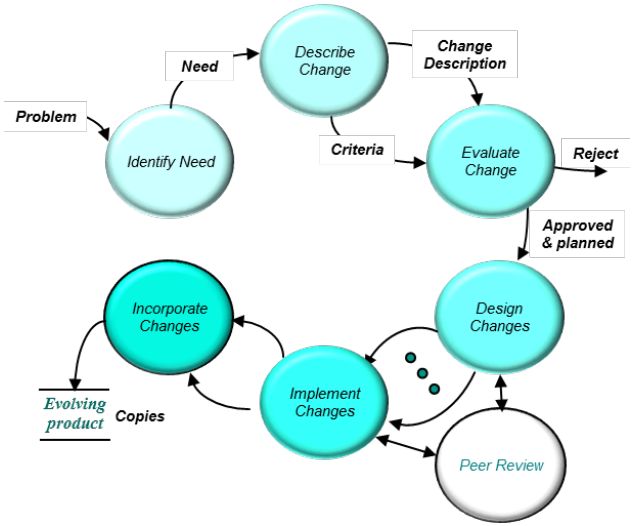


Figure 26-4. Change control process.

- Identifying the need for change can have multiple sources during product development: testing, reviews (verification and validation), integration, additional insight or understanding of the requirement, or altered expectations.
  - Preparing a description of the change should identify: value of the change, which baselines are affected (CIs, relationship data, documents, data, etc.), the estimated extent of the change and impacts. The description of the change for evaluation includes the following information:
    - originator
    - change number
    - funding: planned work breakdown structure identifier (WBS ID) or unplanned (approval)
    - authorization
    - date opened
    - status
    - implementation responsibility
    - implementation approach
    - priority and severity
    - resources (e.g., organization, activity, and phase)
    - size (planned and actual) (e.g., new, changed, deleted)
    - effort (planned and actual)

    - schedule checkpoints (e.g., system requirements, component requirements, design, implementation, inspection, test, integration, regression, system test)
    - internal/external impacts (name, version)
    - related changes (cross reference change numbers)
    - date closed (planned and actual)
  - Evaluation of changes must be responsive yet deliberate. Proposed changes are documented and authorized before being made. All changes must be approved by an appropriate review group. The appropriate level is the highest level representing all impacted stakeholders. Depending on where and when the change is identified, an internal change control board may be a single project team leader or a group of people representing other interfacing CIs. If a required change is identified during integration with other CIs, an internal change control board reviews and approves the modifications.
- Establishing a review board includes determining who has the responsibility for capturing, tracking, assembling, and delivering the actual product, and who has authority to control what happens to completed parts of the product. Review boards are the human component of the CM discipline. More than one board may enact multiple levels of product management (e.g., whether a change should be made, under what conditions it can be released); however, there is one review board that interfaces with the customer, see Figure 26-5.

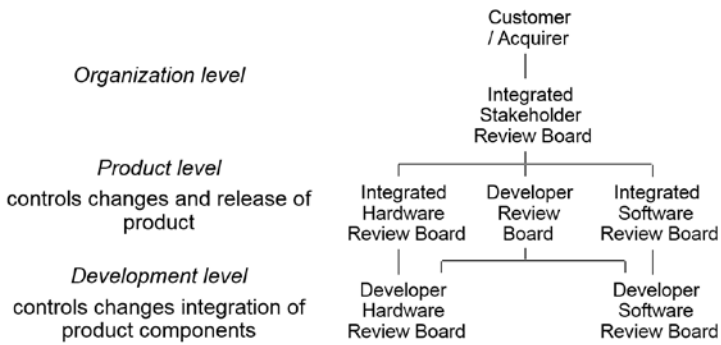


Figure 26-5. Multiple review boards.

- Review board members must be representative of the stakeholders affected by the change and must be knowledgeable about the product. Memberships may change throughout the lifecycle and should be augmented with subject matter experts if needed. Typical review board representatives include:

Hardware Review Board	Software Review Board
Program manager	Product software manager
Configuration management	Software configuration management
Quality assurance	Software quality assurance
Reliability/maintainability	Software systems engineering
Manufacturing (all)	Software engineering
Systems engineer lead	Verification and validation
Product support	Operation/maintenance (if required)

- Questions review boards want answered include the following:
  - Was a copy of the problem report submitted?
  - Is there a complete definition of the problem to be fixed?
  - What are the conditions of the observed problem occurred?
  - Did any requirements change?
  - Which CIs and documents are affected?
  - What are the alternatives to making the change?
  - Is the change within a single component or does it involve others?
  - How many new/changed lines will be required?
  - Is there more than one way to do it?
  - Why was the proposed approach selected?
  - What are the required and planned dates?
  - What are the future product consequences?
  - What are the potential costs/savings for making this change?
  - How critical is the change?
  - Will another change supersede or invalidate this one, or does it depend on other changes?
  - Are there any special test requirements?
  - Are the needed people available to do the work?
  - What are the memory, performance or other system consequences of the change?
  - Are there special advantages to be derived from the change?
  - Are there special considerations such as who is requesting the change or who will it affect?
  - At what point does the change enter the process?
  - How long has the change been under consideration?
  - How much time will this take?
  - Is this rework?
  - Do the corrections properly implement the change?
- Evaluation considerations include impacts to one or more of the following:

- Technical requirements – performance (e.g., size, timing), functionality, external interfaces, safety, user skills, staffing, training, or human engineering
- Non-technical requirements – price, incentives, cost, schedules, guarantees
- Other itemized factors – Government furnished property (GFP), operational, maintenance, test programs, compatibility with support resources, delivered operational documents, operations limits or performance constraints
- Levels of concern – need to renegotiate schedule, cost, or product definition with customer

- The evaluation may result in the decision to approve and schedule now, to plan resource and schedule changes for all affected baselines at a later time with a revisit date, or not to make the change.
  - Design change is supported by locating the correct version of each affected component. All changes made after the design is baselined must be documented and reviewed for incorporation into the design. Designing the modification may require changes to multiple life cycle activities for the impacted products
    - The records create a history of the evolution of the design, which can be invaluable for failure investigation and for facilitating the design of future similar products. These records can prevent the repetition of errors, the development of ineffective designs, and can be invaluable for failure investigation and for facilitating the design of future similar products.
    - Changes to the design are reviewed, validated (or verified where appropriate), and approved.

      Note: The complexity of the modification and its impact to its interfaces may require a thorough peer review of design.

  - Implement changes is supported by ensuring copies of the baselined products are available for modification, so the change can be backed out, if needed, from all representations. As changes are completed, the revised products must meet the same criteria for resubmission to the baseline as original work products. Technical reviews and testing are performed to verify the appropriate changes have been made to all affected components and test results (e.g., unit, integration, regression) are evaluated to validate against approved change description.
  - Incorporate changes is supported by ensuring baselines for all affected components and associated documentation are generated for the evolving product at the scheduled activity checkpoints (e.g., architecture, design, implementation, integration, verification and validation, delivery, deployment).

Status accounting receives information from the other CM functions. The approved configuration documentation, proposed changes and status of implementation, revisions, deviations, and discrepancies are recorded to support traceability and reported. Status accounting systems are closely linked to the data repository used to store and manage the product configuration and product operational information. This allows the status accounting function to collect and tracks data from multiple sources within a project, including:

- Review and audits – in process reviews, functional assessments, physical assessments, supplier compliance.
  - Component tracking
  - Document tracking
  - Control board disposition
  - Baseline reconciliation
  - Product documentation
  - Version description
  - Firmware description
  - Change documents and testing
  - Variances
  - Peer review results
  - Status reports
  - Problem analysis including injection and detection data
  - Corrective action status (i.e., repairs, use-as-is, scrap).
- Product Development File(s) (PDF) are queried to collect pertinent documentation and information. PDFs may be electronic and/or physical (notebook, folder, envelope) files and moves with the product from the time of specification through the baseline process. Data collected in an PDF includes the following:
  - Inputs generated from work products of each development cycle or phase
  - System, database, and interface requirements
  - System architecture
  - System design
  - Internal and external dependencies
  - Test plan, procedures, and reports.
- Baseline data is collected to include the following:
  - Contents (which products)
  - Status (pending update)
  - Revision and distribution history
  - Baseline delta: all changes that have been tested and are ready for incorporation in next baseline
  - Configuration audit results and final disposition of identified discrepancies.

- Product status logs include the following:
  - Change identification
  - Developers
  - Products affected
  - Documentation (documents, pages, changes)
  - Reason(s) for change (change request number)
  - Change dependencies
  - Test runs and results.
- Documentation status logs include the following:
  - Document status
  - List of all product requirements and design documents, drawings
  - Records of all project documents
  - Records of planned changes
  - History of review board decisions
  - History of implemented changes
  - Identification of version/revision compatibility.
- CM performance data
  - Number of open changes and variances awaiting customer disposition
  - Number of closed changes and variances
  - Number of minor changes/variances
  - Number of major changes/variances
  - Change processing cycle time
  - Average amount of time (days) changes or variances remain open
  - Percentage of requests for change/variance by category, nomenclature, CI identifier number, etc.
  - Change request rejection rate and reason for rejection.
- CM controls the ability to access CIs to make authorized changes to system hardware, and software, environment(s), component management, build management, test support (e.g., unit, component, integration, and system), baseline duplication, and disaster recovery. Who has authorization, when can the changes be made, and when will the changes be incorporated into baselines is enforced using tools.
- Configuration verification and audit consists of two components. Verification processes are performed to verify the initial configuration of a CI and the incorporation of approved changes to ensure the CI meets its performance requirements and the processes are common to configuration management, systems engineering, design engineering, manufacturing, and quality assurance. Audit processes ensure the functional and a physical attributes of the contractor's products and associated documentation or system being audited is consistent with the product meeting its requirements.
  - The functional aspect of configuration verification encompasses all of the test and demonstrations performed to meet the quality assurance sections of the applicable performance specifications. The

tests include verification/qualification tests performed on a selected unit or units of the CI, and repetitive acceptance testing performed on each deliverable CI, or on a sampling from each lot of CIs, as applicable. The physical aspect of configuration verification establishes that the as-built configuration is in conformance with the as-designed configuration. The contractor accomplishes this verification by physical inspection, process control, or a combination of both.

- Using the following data:
  - Configuration, status, and schedule information from status accounting
  - Approved configuration documentation from the identification process
  - Results of testing and verification activities
  - Physical hardware CI or software computer software configuration item (CSCI) and its representation
  - Manufacturing
  - Build instructions and engineering tools, including the software engineering environment, used to develop, produce, test and verify the products.
- Data management – data accession, document coordination and tracking, release and distribution, configuration indices, media storage. Its scope includes project data (e.g., plans, processes, procedures, and reports) and technical data, including:
  - product definition information required to define and document an engineering design or product configuration (sufficient to allow duplication of the original items) and is used to support production, engineering, and logistics activities
  - product operational information that provides instructions for the installation, operation, maintenance, training, and support of a system or equipment can be formatted into a technical manual.
  - The data management process assumes all data progresses through four status levels prior to retirement and archival, see Figure 26-6.
- A documentation schema is used to improve accessibility and usefulness with meaningful groupings, see Figure 26-7. This tree is structured by dependence with lower products developed later than higher products, and are dependent on their content.

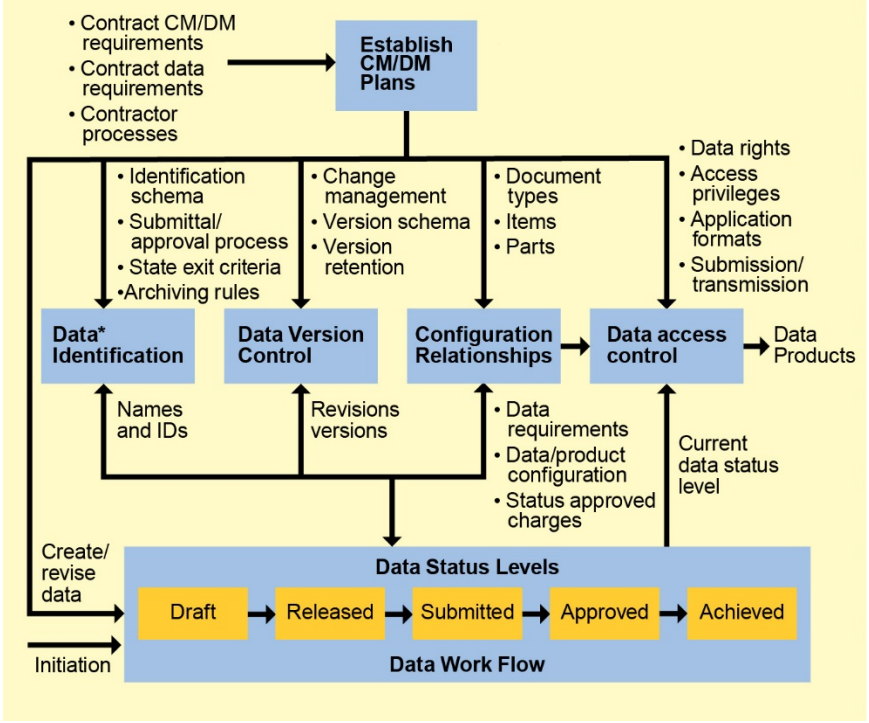


Figure 26-6. Data management processes.

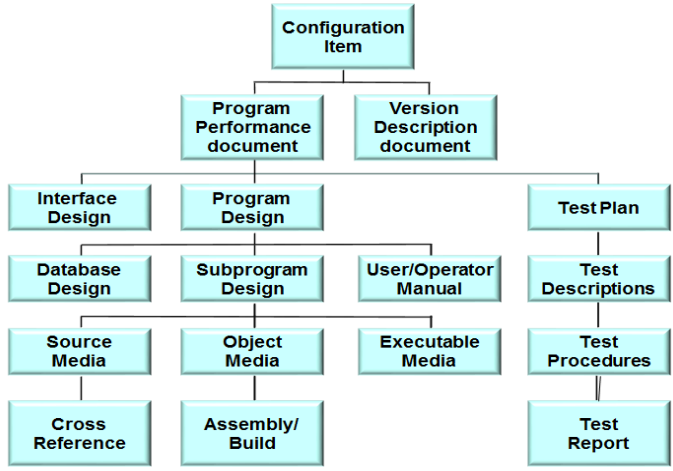


Figure 26-7. Documentation schema.

- Transfer/release processes are used when program/product data changes ownership. The attributes recorded during the process include: item number (ID number, version, and revision), reference item, and document/file description. Supplementary data that may be recorded include a description of the next higher ID number, the baseline configuration, and development/maintenance notes.
- Document evaluation processes include specific criteria to be satisfied prior to establishing the document baseline, these include:
  - Internal consistency with no contradictory statements, consistent definitions
  - Acronyms are defined at first use and included in the glossary
  - Names and descriptions are consistent
  - Understandability that leaves no shadow of doubt as to what is being said, defined or illustrated.
  - Traceability can be tracked upward to predecessor documents and downward to successor documents.
  - External consistency with no contradiction between documents or interfaces
  - Appropriate analysis, design, and development methods and procedures (see SEMP, HDP, SDP)
  - Testability of requirements (i.e., requirement can be tested and the test is feasible/cost effective)
  - Consistency between data definition and data use
  - Appropriate allocation of sizing and timing resources (see SEMP, HDP, SDP, and specifications)
  - Adequate test coverage of requirements
  - Completeness of testing/retesting
  - Adequate quality factors

26.7 Government and Contractor Enabling Processes and Products

The government and contractor teams require CM and DM resources. These resources include personnel, tailored processes, funding, tools, and a computing environment for implementation.

Both the government and the contractor are required to have a CM as well as a DM plan in place before the technical baseline is established. Detailed operating procedures are also necessary in order to implement the plans.

In addition, there are many software tools and products available to implement CM and DM. Some of the more popular CM tools are listed below:

- VSS – Visual source safe
- CVS- Concurrent version system

- Rational Clear Case
- SVN- Subversion
- Perforce

Regarding DM software tools, there is a wide range from which to select. The most prevalent ones in use in the government industry are Livelink and TopVue Defense.

26.8 Configuration Management Examples

26.8.1 Example Subtask Summary Chart for Configuration Management

Phase	Government Enabling Products	Contractor Enabling Products
Phase 0	Identify Government needs for visibility into the contractor's processes and information needed for decision making. Consider contractual requirements for the contractor and those that must be flowed down to the subcontractors and suppliers in statement of work (SOW)/PWS, CDRL/data item descriptions (DIDs), work breakdown structure (WBS) Ensure that all contractor tasks and deliverables are included, that Section M evaluation criteria consider CM and that CM is adequately addressed by the requirements	Provide configuration and data management-related requirements, tasks and deliverables for contractual documents (RFP, SOW/PWS, CDRL/DIDs)
Phase A	Define criteria for integrated baseline review (IBR) in accordance with (IAW) contract. Participate and assess criteria and CDRL satisfaction. Ensure the following are addressed in the configuration and data management planning documents (e.g., SEMP, SDP, CMP, etc.): Processes, procedures, work instructions provide sufficient detail to be repeatable Work product(s) standards and requirements Measurement objectives and needed results (e.g., quality, time scale, cycle time, use of resources) Tasks, work product(s) and data dependencies Facilities, funding, personnel, and tools Roles, responsibility, and authority assignments	Feedback on configuration and data management-related CDRLs, planned baselines, drawings, hardware (including commercial off-the-shelf [COTS]), test data, action item closure, status accounting and reports, program schedules

Phase	Government Enabling Products	Contractor Enabling Products
	Training (role-based and awareness) Work products and data to be controlled by phase are identified, the level of control to be applied, the owners by phase, and when control is established Stakeholders (e.g., who participates in the internal engineering review board (ERB) vs change control board (CCB)) and their participation are identified Monitoring progress and controlling the performance Objective evaluation of performance against plans and processes Management review activities of the process performance and the work product compliance to standards	
Phase B	Identify criteria for preliminary design review (PDR) IAW contract, schedule and plans, e.g., change control of requirements, project data, functional and allocated baselines established, CDRL(s) delivery, CM status and reports, all configuration items are identified. Participate and assess criteria	Feedback/Action Item(s) on configuration and data management tasks, e.g., functional baseline audit results, change control of project plans, trade study results, cross-reference matrices, CDRLs delivered, etc.
Phase C	Review CM entrance and exit criteria for CDR IAW contract, schedule and plans, e.g., configuration control process is executed and maintained, identify key CCB issues, evaluate deviations from CM procedures, requirements changes are controlled, baselines established, CDRLs delivered, CM status and reports are timely. Ensure that the CDR agenda addresses CM and CM deviations. Evaluate CM CDRLs Participate and assess criteria	Feedback/Action Item(s) on configuration and data management tasks, e.g., allocated baseline audit results, preliminary design baselines established,
Phase D1	Review CM entrance and exit criteria for test readiness review (TRR), functional configuration audit (FCA) and PCA IAW contract, schedule and plans. Ensure that all system elements and component are built from authorized baselines, tested, verified and integrated according to qualified MA processes. Assess the verification and validation of manufacturing and integration, test, and evaluation (IT&E), the flight readiness certification and performance risk; Ensure that the configuration control process is appropriately executed and	Feedback on CDRLs (e.g., end item packages, technical reviews, HARS, requirements, quality assurance (QA)/CM reports, concepts of operation (CONOPs), program schedules), configuration and test board materials. Independent readiness, maturity, risk, realism assessment, as appropriate.

Phase	Government Enabling Products	Contractor Enabling Products
	maintained, and that CM issues are appropriated resolved. Evaluate CM CDRLs. Identify CM-related risks to mission performance, reliability, suitability and operability. Ensure that HEO altitude reference system (HARs) have appropriate entrance and exit criteria for CM, and the criteria are satisfied.	
Phase D2	Define Criteria for SVR, MRR, LRR, FRR and independent readiness review team (IRRT); Participate and Assess Criteria and CDRL Satisfaction; Ensure that the configuration control process is appropriately designed, used and maintained. Review the key CM Board issues and evaluate deviations from CM procedures. Ensure that the CDR agenda addresses CM and CM deviations. Review TRR, FCA/PCA, formal qualification review (FQR), HAR and program requirements review (PRR) entrance and exit criteria for CM. Evaluate CM CDRLs. Support IRRT CM activities.	Completion of technical reviews/CDRLS Feedback on IT&E-related CDRLs, requirements, QA/CM reports, CONOPs, program schedules, configuration and test board materials. Supporting documentation to the IRRT and other reviews. Independent IT&E readiness/maturity/risk/realism assessment, as appropriate.
Phase D3	Ensure that ground segment and flight software adhere to appropriate CM practices during operations and support. As appropriate, provide CM assessments and guidance for ground segment and flight software contract changes (upgrades, block changes), study efforts and routine operational patches.	Repeat the appropriate tasks to support the ground segment and flight software upgrades during operations

26.8.2 Example CM Processes for Configuration Management

26.8.2.1 CM Planning and Management

Entry	Inputs	Activities	Outputs	Exit
Program initiation Management support Team interfaces Facilities and Tools Training requirements	Contract requirements Standard processes and procedures with tailoring guidance Program plans Resources Time	Identify personnel Establish subcontractor/vendor CM requirements Document roles and responsibilities Establish environment: facilities and tools. Plan training: awareness and skills building Develop meeting and reporting requirements Tailor processes, procedures, and standards for: Identification naming and numbering conventions Baseline requirements Change control Status accounting Audits and reviews Data management Identify what and when work products will be placed under CM (e.g., user stories, requirements, architecture documentation, design data, breadboards, prototypes, tools, tool configurations, drawings, specifications, hardware and equipment, software tools and libraries, compilers, test tools and test scripts, data, acquired products, product data files, product technical documents, products delivered to customers, installation logs, iteration backlogs, program plans, specific internal work products: processes, procedures, and other items used in creating and describing these work products)	CM Program Plan Team members with assigned CM roles and responsibility	Management approved of CM: Process tailoring Processes Plans Resources Schedules

26.9 Bibliography

Specifications and Standards

EIA-649-1 Configuration Management Requirements for Defense Contracts, Nov 14

MIL-STD-2549 Configuration Management Data Interface, 30 June 97

MIL-STD-31000 Technical Data Packages

Technical Handbooks

MIL-HDBK-61A, Configuration Management Guidance, 7 Feb 01

Contractor Deliverables

DI-ALSS-81535 Deficiency Report (DR)

DI-CMAN-80639 Engineering Change Proposal (ECP)

DI-CMAN-80642 Notice of Revision (NOR)

DI-CMAN-80643 Specification Change Notice (SCN)

DI-CMAN-80858 Contractor's Configuration Management Plan

DI-CMAN-81551 Drawings, Specifications, Standards, Software, and Software Support Documents Data Information Packet

DI-IPSC-81442 Software Version Description

DI-MGMT-80368 Status Report

DI-MGMT-80377 Government Furnished Equipment Detail Transaction Status Data

DI-MGMT-81024 System Engineering Management Plan

DI-MGMT-81232 System Problem Report (SPR)

DI-MGMT-81453 Data Accession List (DAL)

DI-MISC-81454 Automated Computer Program Identification Number (ACPIN) Data and Control Record

DI-MISC-81807 Software/Firmware Change Request

DI-SESS-80776 Technical Data Package

Other

AFI 63-131, *Modification Management*, 19 Mar 13

AFI 10-601, Operational Capability Requirements Development, 6 Nov 13

SMCI 62-109, SMC *Configuration Management Process*, DRAFT, 11 Jun 14

SMC-S-002, Configuration Management, 13 Jun 08

TOR-2006(8583)-1 *Configuration Management* 15 Aug 05

26.10 Acronyms

ABL	allocated baseline
ACPIN	automated computer program identifier number
CAD	computer-aided design
CAGE	commercial and government entity
CCB	configuration control board
CCB	change control board
CDRL	contract data requirements list
CI	configuration item
CM	configuration management
CONOPS	concepts of operation
COTS	commercial off-the-shelf
CSCI	computer software configuration team
CVS	current version system
DAL	data accession list
DIDs	data item descriptions
DM	data management
DUNS	Data Universal Numbering System
ECP	engineering change proposal
ERB	engineering review board
FCA	functional configuration audit
FQR	formal qualification review
GEIA	Government Electronics and Information Technology Association

GFP	government-furnished property
HARS	HEO attitude reference system
HDP	hardware development plan
HW	hardware
IAW	in accordance with
IBR	integrated baseline review
ICD	initial capabilities document
IPT	integrated product team
IPT	integrated product team
IRRT	independent readiness review team
ISO	International Standards Organization
IT&E	integration, test, and evaluation
NOR	notice of revision
PBL	product baseline
PCA	product configuration audit
PDF	product development file
PDR	preliminary design review
PM	program manager
PROM	programmable read-only memory (?)
PRR	program requirements review
QA	quality assurance
SCN	specification change notice
SDP	software development plan
SEMP	systems engineering management plan
SOW	statement of work
SPO	space program operations
SPR	system problem report
SVN	subversion
SW	software
TRR	test readiness review
VSS	visual source safe
WBS	work breakdown structure
WBSID	work breakdown structure identifier

Chapter 27
Metrics

Gary C. Palosaari and Daniel X. Houston
Software Acquisition and Modeling Department
Software Engineering Subdivision

27.1 Introduction

The government often calls upon its technical partner to provide objective information based on measurements taken on programs and products. Responding to government requests requires knowledge of measurement processes and tools as well as skills in selecting and specifying measures; implementing measures; collecting and analyzing data; and producing, interpreting, and reporting metrics. Although an acquisition partner is rarely called upon to oversee or guide an entire measurement process for a program, knowledge of the entire process is required to report metrics with proper qualifications and to use metrics with integrity.

Metrics are an important part of any ground system development effort. They can be used to help with cost estimates, to help manage development and sustainment, to help assess product quality, and to understand system characteristics and performance. Measurement by itself cannot solve problems, but it can clarify and focus understanding of them. Contractors and system program offices (SPOs) have different areas of focus regarding system development. Contractors are primarily concerned about open-ended requirements, uncontrolled changes, insufficient testing, inadequate training, arbitrary schedules, insufficient funding, and issues related to standards, product reliability, and product suitability. A SPO is primarily concerned with delivered functionality and product quality, not day-to-day management of product development. Thus the SPO may focus on a subset of an entire set of metrics that are collected and analyzed by the contractor. In addition, a SPO may need to collect and analyze additional metrics not provided by the contractor.

27.2 Definitions

Metrics and measures are often used synonymously, but they have different meanings. In the context of this discussion we use the following definitions.

Measure A quantitative indication of some attribute of a system or process. For example, source lines of code (SLOC) is a measure used to determine the size of a software system.

Metric A quantitative measure of the degree to which a system or process possesses an attribute [1]. For example, a software system that has 10,000 SLOCs is much smaller than a software system that has 1 million SLOCs.

A few key measurement definitions are listed here. For a more complete set of measurement terminology, see either *Software Measurement Standard for Space Systems* TOR-2009(8506)-6, [2] or *Practical Software and Systems Measurement* [3].

Attribute A property or characteristic of an entity that can be distinguished quantitatively or qualitatively by human or automated means. An attribute is the property or characteristic of an entity that is quantified to obtain a base measure.

Base measure A measure of a single attribute defined by a specified measurement method. A base measure is functionally independent of other measures.

Derived measure A quantity that is defined as a function of two or more values of base and/or derived measures.

Information need What the measurement user (e.g., manager or project team member) needs to know in order to make informed decisions.

Measureable concept An abstract relationship between attributes of entities and information needs, sometimes documented as a question that addresses an information need.

Unit of measurement (UOM) The standardized, quantitative amount that will be counted to derive the value of the base measure, such as an hour or line of code. A unit is a particular quantity, defined and adopted by convention, with which other quantities of the same kind are compared in order to express their magnitude relative to that quantity.

27.3 Objectives

Measurement enables us to live with a sense of predictability and control. For example, alarm clocks help us rise in time to meet scheduled activities, fuel gauges help avoid running out of gas, and speedometers help us maintain safe driving speeds. For more complex activities, such as systems development and sustainment, Fenton and Pfleeger point out that we have additional reasons for measuring: understanding and improvement [4]. Both acquisition programs and the systems they produce often behave in ways that we do not understand, so measurements are taken to provide insight to behavior. When observed behavior does not match desired behavior, improvements may be sought and measurements can provide feedback on the success of the improvement effort.

Gaffney et al. [5] offer a further decomposition of measurement objectives with a list of five motivations for measurement:

- Measurement enables an organization to communicate effectively by providing objective information.
- Status measures track progress against specific program objectives.
- Measures can be used to identify problems early and to correct them.
- Management needs to understand trade-offs prior to making decisions. Measures provide information about potential impacts.
- Measurement provides an effective rationale for estimating and planning, for selecting among alternatives, and for justifying and defending decisions.

27.4 Practices

27.4.1 Core Activities

Measurement in the context of a development or sustainment program is usually performed for the sake of the managing the program and product, as opposed to supporting a specific decision. The following activities are key measurement tasks for program and product management.

- *Estimation*
  Prior to measuring program actual data for the sake of ongoing management, estimates are produced for planning and allocating resources. Estimation is based on knowledge of performance on prior programs and on knowledge of the characteristics of the program being estimated. Many estimation methods and tools are available. Some are general methods, such as expert judgment, estimation by analogy, and regression analysis. However, these methods are applied to specific disciplines in tools that incorporate field-specific data.

  A few estimations guidelines can be offered [6].
  - When preparing estimates, data from previous projects is preferred over expert judgment without data. As for the type of data, an organization's own data from previous projects is better than industry data, including databases that are contained in commercial estimation tools.
  - Estimation accuracy can increase when multiple estimation methods are used because some estimates may cluster while others are outliers. But also, multiple estimates lead to questions about the accuracy of inputs and testing how sensitive an estimate maybe to its inputs.
  - As a project proceeds, refine estimates using actual results.

- *Assure a measurement program and plan exists*
  A Measurement and Analysis Plan (MAP) may be developed to cover the system and software measurement activities for all team members. When available, the MAP should define applicable measurement procedures and identify who will be responsible for their performance. The MAP should also define the integration of sub-contractor/vendor activities. Existing policies/procedures should be referenced within the MAP and be available for review by the government, as required. Measurement milestones and periodic reviews (e.g. design reviews, program management reviews, etc.) should be identified at the respective level of the program schedule for visibility and monitoring by the SPO. If a MAP covering both systems and software metrics is not developed, then required measures should be identified in a System Engineering Plan (SEP)/System Engineering Management Plan (SEMP) and a Software Development Plan (SDP).

  As discussed in the *System Engineering Primer and Handbook*, the system engineering processes and responsibilities for implementation are usually described in a SEP for the program [7]. Its counterpart, the SEMP is typically produced by the contractor with corresponding information. Per the *System Engineering Leading Indicators Guide* the SEP discusses the program's approach for using metrics to monitor prime and subcontractor contract execution for impacts to cost, schedule, and performance (i.e., software deficiency reports; technical performance measures (TPMs); Critical Technical Parameters (CTPs); measures of effectiveness (MOEs); measures of suitability (MOSs); measures of performance (MOPs); and Sustainment key performance parameters (KPP)/key system attributes (KSAs) (i.e., materiel availability, materiel reliability, ownership cost, and mean down time) [8]. The metrics provider and reporting frequency are also identified.

  In the software domain, a Software Measurement Plan (SMP) addresses software specific metrics. The SMP provides the planned metrics and their aggregation levels, explanations of computations, expected or projected values, thresholds, and any planned corrective actions to be taken in case thresholds are breached. A template for the SMP is provided in the *Software Development Standard for Mission Critical Systems* [9]. When a separate SMP is not required on contract, the SDP should include the content described in the SMP template.

- *Select measures*
  The Goal-Question-Metric (GQM) method should be used to translate program goals into questions that are then associated with measures [10, 11]. For example, a system engineering goal may be to establish a

stable, complete requirements set. A corresponding question is "Is the system engineering effort driving towards stability in terms of the system definition (and size)?" One of the associated measures is "Percent of Requirements Approved". Guidance for selecting the measures based on priority, feasibility, lifecycle phase etc. is provided in the international standard, ISO/IEC 15939, *Systems and Software Engineering – Measurement Process - Annex C* [12].

An example showing a complete mapping of sustainment program software goals to questions and measures is provided in *Recommended Software Measures for MILSATCOM Sustainment Programs* reproduced in Table 27-1 for reference [13].

Table 27-1. Software Sustainment Goals, Questions, and Measures

Goal	Question	Measure
Maximize Customer Satisfaction	What is the source of the problem reports? What portion is due to bad fixes? To documentation?	Software Problem Reports by Source
	How do software releases compare in the number of problem reports fixed and the effort applied? What is the average effort per delivered fix?	Software Problem Report Backlog Software Release Comparison
	Have many software problems been found relative to the software size?	Discovered Defect Density
	Are software problem reports being addressed in a timely manner? Are the reports being addressed in order of priority?	Software Problem Report Time in Process (TiP) by Category Software Problem Reports Aging by Category
	Are software problem reports being worked in a timely manner?	Software Problem Report Time in Process (TiP) by Category
	How large is the backlog of problem reports?	Software Problem Report Backlog

Goal	Question	Measure
	What is the current capability of the sustainment process? Can it keep up with the workload?	Backlog Management Index (BMI)
	Are releases being delivered as planned?	Software Release Schedule Delay
	Are there many bad fixes?	Fix Integrity
Minimize Cost	Is the software growing significantly through sustainment?	Software Size Change by Release
	What has been the staffing level and how has it changed during the program?	Software Sustainment Staff Level
	Are software problem reports being addressed in a timely manner? Are the reports being addressed in order	Software Problem Reports Aging by Category
Minimize Duration	Are releases being delivered as planned?	Software Release Schedule Delay
	How do software releases compare in the number of problem reports fixed and the effort applied? What is the average effort per delivered fix?	Software Release Comparison

- *Specify measures*
  Establishing precise, consistent definitions in a measurement specification document is an essential part of any measurement program. The results of the GQM measurement selection exercise provide a starting point for the specification process. Measurement specifications should include information needs, measureable concepts or questions, measurement methods, units of measure, samples, thresholds, decision criteria etc. *Practical Software and Systems Measurement* [3], sponsored by the Department of Defense and the US army, provides methods and templates to specify measures per ISO/IEC 15939. As part of this process, the information needs should be identified and prioritized, based on goals/objectives, risks, known problems, and improvement opportunities. Useful guidance on risks that may provide a source for information needs can be found in ISO/IEC 16085:2006. Note that in the process of specifying measures, it may be useful to iterate back to the GQM steps, since information needs are associated with project goals, and information needs can be posed as questions.

  The *Software Measurement Standard for Space Systems* provides a starting point for software measurement with measurement specifications based on the PSM [2]. These specifications include descriptions of base measures, which provide a measure of a single attribute and are functionally independent of other measures. Derived measures, which are a function of two or more values of base and/or

derived measures are another key component of the measurement specifications. The derived measures specified by Abelson et al. are grouped in a structured list or taxonomy by Information Categories and Management Indicators/Measurement Concepts (see Table 27-2) [2]. This set of measures is a comprehensive and balanced list (reference Section 3.5, Key Lessons Learned). The measurement specifications defined in Abelson et al. also identify reporting frequency and suggested thresholds for taking action [2]. The SPO may want to add or subtract from the recommended list of measures over time, depending on current risk areas and the lifecycle phase. Specific issues such as ground system usability can be measured through the Discrepancy Report by Type Indicator with the addition of a usability data type, which includes readability. Note that another key ground system operator measure is the Response Time Indicator, which captures the elapsed time from a scenario stimulus to the beginning of the system's response.

Table 27-2. Software Measurement Set

Information Categories	Management Indicators	Measurements
Schedule and Progress		Requirements defined
	Requirement	Requirements TBX closure
	Progress	Requirements verified
		Qualification method
		Components defined
	Development	Units defined
	Progress	Units coded and unit tested
		Units integrated and tested
		Test cases developed
	Test	Test cases dry run
	Progress	Test cases performed
		Test cases passed
	Schedule	Project milestones
	Adherence	Scheduled activities
Resources and Cost	Effort	Labor hours by activity
	Profile	Rework hours by activity
	Staff	Staffing level
	Profile	Staff experience

Information Categories	Management Indicators	Measurements
		Staffing turnover
		CPU utilization
	Computer	Memory utilization
	Resource	Input/output utilization
		Response time
		Earned value performance
	Cost Profile	Schedule and cost performance index
		Schedule and cost variance
Product Size and Stability		Requirement size
		Requirements by type
	Size	Line of code size
		Line of code by origin
		Line of code by type
	Volatility	Requirement volatility
		Line of code volatility
	Build Content	Requirements per build
Product Quality	Discrepancy	Discrepancy report status
	Report	Discrepancy report aging
	Resolution	Discrepancy report by type
		Discrepancy report by source
	Complexity	Cyclomatic complexity
	Coverage	Requirements to design traceability
		Requirements to test case traceability
Development Performance	Productivity	Development productivity
	Maturity	Development defect density
	Management	Action item closure
	Status	Risk mitigation task status
		Schedule compression

In the systems engineering domain, a team composed of industry, government and academia measurement experts in collaboration with the International Council on Systems Engineering (INCOSE) has applied the PSM specification methods to identify eighteen systems engineering leading indicators System Engineering Leading Indicators Guide [14]. These leading indicators are associated with multiple measures that can be used to provide trend information for predictive analysis. Table 27-3 Systems Engineering Leading Indicators provides a summary showing the leading indicators along with the insight provided. The measurement specifications for each of the system engineering

indicators are also provided in [14]. These specifications list base measures and multiple derived measures for each indicator, along with graphical examples and other specification parameters identified in the PSM methodology.

Table 27-3. Systems Engineering Leading Indicators

Leading Indicator	Insight Provided
Requirements Trends	Rate of maturity of the system definition against the plan. Additionally, characterizes the stability and completeness of the system requirements that could potentially impact design, production, operational utility, or support.
System Definition Change Backlog Trend	Change request backlog which, when excessive, could have adverse impact on the technical, cost and schedule baselines.
Interface Trends	Interface specification closure against plan. Lack of timely closure could pose adverse impact to system architecture, design, implementation and/or V&V any of which could pose technical, cost and schedule impact.
Requirements Validation Trends	Progress against plan in assuring that the customer requirements are valid and properly understood. Adverse trends would pose impacts to system design activity with corresponding impacts to technical, cost and schedule baselines and customer satisfaction.
Requirements Verification Trends	Progress against plan in verifying that the design meets the specified requirements. Adverse trends would indicate inadequate design and rework that could impact technical, cost and schedule baselines. Also, potential adverse operational effectiveness of the system.
Work Product Approval Trends	Adequacy of internal processes for the work being performed and also the adequacy of the document review process, both internal and external to the organization. High reject count would suggest poor quality work or a poor document review process each of which could have adverse cost, schedule and customer satisfaction impact.
Review Action Closure Trends	Responsiveness of the organization in closing post-review actions. Adverse trends could forecast potential technical, cost and schedule baseline issues.

Leading Indicator	Insight Provided
Risk Exposure Trends	Effectiveness of risk management process in managing/ mitigating technical, cost and schedule risks. An effective risk handing process will lower risk exposure trends.
Risk Treatment Trends	Effectiveness of the systems engineering (SE) organization in implementing risk mitigation activities. If the SE organization is not retiring risk in a timely manner, additional resources can be allocated before
Technology Maturity Trends	Risk associated with incorporation of new technology or failure to refresh dated technology. Adoption of immature technology could introduce significant risk during development while failure to refresh dates technology could have operational effectiveness/ customer satisfaction impact.
Technical Measurement Trends	Progress towards meeting the Measures of Effectiveness (MOEs)/Performance (MOPs)/KPPs and Technical Performance Measures (TPMs). Lack of timely closure is an indicator of performance deficiencies in the product design and/or project team's performance.
Systems Engineering Staffing & Skills Trends	Quantity and quality of SE personnel assigned, the skill and seniority mix, and the time phasing of their application throughout the project lifecycle.
Process Compliance Trends	Quality and consistency of the project defined SE process as documented in SEP/SEMP. Poor/inconsistent SE processes and/or failure to adhere to SEP/SEMP, increase project risk.
Facility and Equipment Availability Trends	Availability of non-personnel resources (infrastructure, capital assets, etc.) needed throughout the project lifecycle.
Defect/Error Trends	Progress towards the creation of a product or the delivery of a service that meets the quality expectations of its recipient. Understanding the proportion of defects being found and opportunities for finding defects at each stage of the development process of a product or the execution of a service.
System Affordability Trends	Progress towards a system that is affordable for the stakeholders. Understanding the balance between performance, cost, and schedule and the associated confidence or risk.

Leading Indicator	Insight Provided
Architecture Trends	Maturity of an organization with regards to implementation and deployment of an architecture process that is based on an accept set of industry standards and guidelines.
Schedule and Cost Pressure	Impact of schedule and cost challenges on carrying out a project

- *Implement measures*
  To implement the measurement process, the contractor team must be trained on data collection and storage procedures along with the use of data collection tools and analysis methods. The base measurement data should be collected and stored for easy access. Integrity checks should be performed to identify measurement process errors and missing or inconsistent data. Finally, derived measures should be calculated from the base measures per the definitions in the measurement specifications.

- *Analyze and report results*
  After collecting and processing measurement data, it must be analyzed and reported to decision makers. Analysis includes interpretation of measurement graphs, often using measurement thresholds (e.g. +-10% of plan) as defined in the measurement specifications. These thresholds are often established at part of contractual negotiations, and should be based on historical data. Example graphs and analysis methods for the each of the recommended software and systems measurement sets (Table 27-2 and Table 27-3) are provided by Abelson [2] and Roedler [14], respectively. Additional guidelines for both systems and software measurement process activities are provided in the Measurement and Analysis process area description provided in the Software Engineering Institute's CMMI® for Development, Version 1.3 [15].

  Measurement results, with analysis and interpretation, should be communicated regularly to relevant stakeholders including contractor management and the SPO to support decision making and corrective action management. For example, if the actual number of test cases performed is being tracked relative to plan with a 10% threshold, then if the threshold is exceeded, management should determine the root cause of the deviation and take action to correct any problems. In the software domain, metrics graphs/results and analysis are typically provided monthly in a Software Measurement Report (SMR), based on a Software Measurement Report (SMR) template [16].

27.4.2 Standards/Recommended Practices

The International Standard ISO/IEC 15939, identifies the activities and tasks that are necessary to successfully identify, define, select, apply and improve measurement [12]. It also provides definitions for measurement, but it does not provide a recommended set of measures to apply on projects.

The *Software Measurement Standard for Space Systems* provides a comprehensive software measurement list and specifications for the analysis and reporting of management indicators [2]. The measures listed in the standard are based on best practices and lessons learned from using software measurements for mission-critical software development projects.

Roedler, et al. have identified systems engineering recommended engineering measures along with the associated measurement specifications in *Systems Engineering Leading Indicators Guide* [14].

The *Software Development Standard for Mission Critical Systems.* Aerospace Report No. TR-RS-2015-00012, March 2014 provides various software measurement templates, including a Software Measurement Plan (SMP) template and a SMR template [9].

27.5 Key Lessons Learned

27.5.1 Balanced Measures

Programs are subject to conflicting constraints: cost, schedule product quality, and product functionality. Many measures are available for these constraints as they are decomposed by phase, system level, activity, work product, and so forth. Similarly, products are subject to conflicting constraints, but the nature of such constraints can vary widely. A few examples are throughput, storage capacity, timing, reliability, and resolution. A program provides many possibilities for measurement. Measures may be chosen because they are easy and because they are important, but one often overlooked criteria is the need for balanced measures.

An example of a lack of balanced measures is a common problem of tracking of cost and schedule on software development projects, while ignoring product quality because software quality measures are more challenging. However, when quality is not measured, undiscovered problems accrue until the need for rework delays progress and increases cost. To prevent cost and schedule overruns, quality measures must be balanced with measures of cost and schedule.

27.5.2 Government Required System/Software Engineering Measures

The ability of a technical partner to support the Government in an acquisition program depends to a high degree on the quality of objective information provided by a contractor. A contractor's measures may not include those that are adequate for Program Office oversight, and there is generally a reluctance to add measures once a program is underway and the Program Office is asking about the status of work and probability of successful completion. Therefore, measures that will enable a program office to advise the Government should be included in the contract. The Software Measurement Standard for Space Systems provides recommended measures for software systems. In addition, the Systems Engineering Leading Indicators Guide provides recommended measures for systems engineering [14].

27.5.3 Base Measures for Analysis and Data Capture

Contractors often have their own non-standard way of measuring that does not provide the SPO insight that they need. Since the SPO often wants to gain additional insight into program performance, base measurements should be obtained from the contractor. Then the data can be independently analyzed, and possibly included in a measurement tool/database like Software Measurement Analysis and Repository Toolset (SMART), an Aerospace proprietary tool that is based on the set of measures and analysis techniques identified by Abelson 2011 [2]. SMART provides an efficient way to store, organize and analyze software measurement data, and it also provides an archive mechanism to capture data for the long term, so that analysis can be performed across programs.

27.5.4 Measures for Risk Assessment

Risks, lifecycle models and processes affect measurement selection. For example, for a program following an agile methodology, the focus on time boxing with agile time periods (e.g., sprint, cycle, release) imply that there needs to be a measure of functionality preceding commitment (e.g. number of story points).

27.5.5 Access to Contractor Measurement Repositories

The existence and location of the program software measurement data libraries/repositories should be defined and accessible by the SPO.

27.5.6 Measures for Decision Making

The need to develop measures for making specific decisions often arises during development of an RFP or a contract change proposal (CCP), or at a critical

point in the program's life cycle. Two examples are the award of contract incentives and forecasting completion of testing.

- *Incentive plans*
  A program may propose an incentive plan to motivate a contractor to meet specific goals, for example, to reduce costs or to produce a reliable system. In order for the program to determine if the contractor is meeting its goals, the goals have to be well defined and measurable, with the best goals based on measurements from the contractor's previous performance. Therefore, data from prior work should be collected and analyzed to establish the goals for the incentive plan. Distributions developed from the data provide a basis for setting compliance thresholds and incentive scales. For example, a common measure of reliability is system availability (A_O). Given a left-skewed distribution of historical annual A_O values, incentive planners may choose a compliance threshold of 20%, meaning a sustainment contractor must achieve an annual A_O value that is at or above the 20^{th} percentile in order to comply with the contract and be eligible for incentive fee. Furthermore, the planners may choose the 50% percentile for the start of the portion of incentive payment for A_O, a payment that increases linearly to a maximum at $A_O = 1$. In this example, historical data collected from past measurements is used to produce scales for judging current measured performance.
- *Forecasting test completion*
  A phenomenon common to large software-intensive systems is a lack of discipline in quality-inducing development activities for the sake of working to a schedule. Work product peer reviews, static analysis, and unit testing may not be performed well or completely, allowing shortcomings to accrue in work products until integration. Upon entering system integration and testing, large number of deficiency reports are found and a system test team can find itself in a seemingly interminable test-and-fix loop. Meanwhile the customer is trying to plan a launch or deployment of a ground system, both expensive operations that depend on timely readiness of the system. Estimates of completion can be produced from integration and test data, but the best way to estimate completion of system testing is to produce a dynamic model of the contractor's test-and-fix process and run it many times to produce a probabilistic estimate of completion.

27.6 Task Execution by Phase

Within the Mission Assurance Verification task database, measurement tasks are first assigned to one of the following seven Mission Assurance Guide (MAG) phases:

- Phase 0: Pre-KDP A Concept Studies
- Phase A: Concept Development
- Phase B: Preliminary Design
- Phase C: Complete Design
- Phase D1: Fabrication and Integration
- Phase D2: Fielding and Checkout
- Phase D3: Operations and Disposal

Table 27-4 tasks ensure that measurement processes are well defined with good estimates, metrics selection, measurement specification, measurement implementation and data collection oversight, data analysis, and metrics interpretation and reporting. The assurance task products are estimates, measurement specifications, data files, and metrics reports. Measurement tasks begin in Phase 0 with estimates developed in concept studies and identification of program goals and risks. These findings are the basis for considering selection of metrics in a GQM exercise as well as understanding information needs to be used later in developing measurement specifications. If contractor performance incentives are to employ measures based on historical data, work in pre-Phase A ensures that the request for proposal (RFP) adequately addresses data needs and analysis for contractor performance incentive measures. During this time, work can begin on choices of program metrics in the statement of work (SOW), contract data requirements list (CDRL), data item descriptions (DIDs), and specifications. As goals, risks, and information needs change between phases, and even within phases, measures must be updated accordingly. The primary focus of measurement during Phases A-D3 is to ensure that metrics are well-chosen, CDRLs and DIDs for measurements are completed accurately, and measurements are adequately addressed at system requirements review (SRR), system design review (SDR), preliminary data review (PDR), and critical design review (CDR) at the system, subsystem, and unit levels.

Table 27-4. Key Measurement Tasks by Phase

Phase	Measurement Tasks
0	• Produce estimates for subsequent phases • Use goals, risks, and information needs to select metrics for each subsequent phase • Specify measures to be taken for phases through the term of the contract • Develop contract items such as SOW, CDRL/DIDs, RFP, work breakdown structure (WBS) to ensure that all contractor measurement tasks and deliverables are included and that evaluation criteria consider measurement • If contractor performance incentives are to be based on measured performance, design the incentive plan and incorporate performance goals

Phase	Measurement Tasks
	based on probabilities derived from historical performance (requires obtaining and analyzing data)
A	• Refine estimates for subsequent phases • Assess contractual implementation of contracted measurement reporting (SOW, CDRL/DIDs, RFP, WBS) to ensure that all contractor tasks and deliverables are included • Interpret derived measures for reporting to customer. As necessary, use base measures to confirm or enhance insights • Review measurable SRR and SDR entrance and exit criteria to ensure that measures are adequately addressed
B	• Refine estimates for subsequent phases • Ensure that all entities and their measured attributes are identified correctly. Review measurable PDR entrance and exit criteria to ensure measures are adequately addressed. • Interpret derived measures for reporting to customer. As necessary, use base measures to confirm or enhance insights • Assess measurable PDA and PDR criteria • Assess measurement CDRLs
C	• Refine estimates for subsequent phases • Review measurable CDR and PRR entrance and exit criteria to ensure that measures are adequately addressed • Interpret derived measures for reporting to customer. As necessary, use base measures to confirm or enhance insights • Ensure that change management measures are implemented • Assess measurement CDRLs
D1	• Refine estimates for subsequent phases • Ensure that the measurement process is appropriately executed and that results are used to identify risks to mission performance, reliability, suitability and operability • Ensure that measurable FCA/PCAs have appropriate entrance and exit criteria and the criteria are satisfied • Assess measurement CDRLs
D2	• Refine estimates for subsequent phases • Ensure that the measurement process is appropriately executed and that results are used to identify risks to mission performance, reliability, suitability and operability • Review measurable entrance and exit criteria for TRR, FCA/PCA, FQR, and PRR • Support IRRT activities with quantitative results • Assess measurement CDRLs

Phase	Measurement Tasks
D3	• As appropriate, provide assessments and guidance for ground segment and flight software contract changes (upgrades, block changes), study efforts and routine operational patches

27.7 Government and Contractor Enabling Processes and Products

In addition to requiring access to the government's draft and final RFP and the negotiated contract, the government's measurement team needs access to the contractor's metrics plan and measurement guidance documentation across the prime and subcontractors. The measurement team will also need access to the contractors' engineering team at all levels of the program through the active design period and testing activities. All data, both collected and analyzed during the program, is also needed. Table 27-5 lists the enabling measurement products for reference.

Table 27-5. Enabling Measurement Products

Phase	Government Enabling Products	Contractor Enabling Products
0	RFP, SOW, CDRL, DIDs, WBS, Estimates for subsequent phases	Proposal, SDP & SMR templates/process descriptions, SEP & SEMP (drafts)
A	Final Contract Criteria for SRR & SDR Refined estimates for subsequent phases	Completion of IBR, SRR, SDR, Measurement Plan, Release Plan, CCB Procedure Base measure data and derived metrics
B	Entrance/Exit Criteria for PDR Refined estimates for subsequent phases	Completion of PDA, PDR Completion of CDRLs Base measure data and derived metrics
C	Entrance/Exit Criteria for CDR and PRR Refined estimates for subsequent phases	Completion of CDA, CDR Completion of CRLS Base measure data and derived metrics
D1	Entrance/Exit Criteria for FCA/PCAs Refined estimates for subsequent phases	End item data packages Completion of CDRLS, FCAs, PCAs Base measure data and derived metrics
D2	Entrance/Exit for Criteria for SVR, MRR, LRR, FRR & IRRT	Completion of readiness reviews

Phase	Government Enabling Products	Contractor Enabling Products
	IRRT results	Base measure data and derived metrics Completion of CDRLS
D3	Government assessments and guidance for Ground Segment software	Contractor ECPs to support the Ground Segment software upgrades during operations

27.8 Practice Measurement Task Application Example

When assessing measurement on a program, the first step is to determine the phase of interest and where in the program office WBS the measurement activities are managed. The appropriate measurement subject matter expert (SME) then assists the program office in determining what measurement tasks are needed using this guide as a roadmap. To assist the program office, a standard reference set of measurement tasks (Table 27-6) can be tailored to the program class (A, B, C, D). The measurement products are then archived over the life cycle. This archive assists in the verification of accomplishment criteria associated with major milestones defined in gated processes such as the Management Plan. It can also be used for future analysis in developing incentive plans (see Section

Table 27-6. Reference Set of Measurement Tasks

	Phase						
Measurement Task	**0**	**A**	**B**	**C**	**D1**	**D2**	**D3**
Produce/refine estimates for subsequent phases	X	X	X	X	X	X	
Use goals, risks, and information needs to select metrics for each subsequent phase	X	X	X	X	X	X	
Specify measures to be taken for phases through the term of the contract	X						
Develop contract items such as SOW, CDRL/DIDs, RFP, WBS to ensure that all contractor measurement tasks and deliverables are included and that evaluation criteria consider measurement	X						
Design the incentive plan. Obtain and analyze data. Incorporate performance goals based on probabilities derived from historical performance.	X						
Assess contractual implementation of contracted measurement reporting (SOW, CDRL/DIDs,		X	X	X	X	X	X

	Phase						
Measurement Task	**0**	**A**	**B**	**C**	**D1**	**D2**	**D3**
RFP, WBS) to ensure that all contractor tasks and deliverables are included							
Interpret derived measures for reporting to customer. As necessary, use base measures to confirm or enhance insights		X	X	X	X	X	X
Review measurable SRR and SDR entrance and exit criteria to ensure that measures are adequately addressed		X					
Ensure that all entities and their measured attributes are identified correctly. Review measurable PDR entrance and exit criteria to ensure measures are adequately addressed.			X				
Assess measurable PDA and PDR criteria			X				
Assess measurement CDRLs			X	X	X	X	X
Review measurable CDR and PRR entrance and exit criteria to ensure that measures are adequately addressed				X			
Ensure that change management measures are implemented				X			
Ensure that the measurement process is appropriately executed and that results are used to identify risks to mission performance, reliability, suitability and operability					X		
Ensure that measurable FCA/PCAs have appropriate entrance and exit criteria and the criteria are satisfied					X		
Ensure that the measurement process is appropriately executed and that results are used to identify risks to mission performance, reliability, suitability and operability						X	
Review measurable entrance and exit criteria for TRR, FCA/PCA, FQR, and PRR						X	
Support IRRT activities with quantitative results						X	
As appropriate, provide assessments and guidance for ground segment and flight software contract changes (upgrades, block changes), study efforts and routine operational patches							X

27.9 References

1. Pressman, Roger. Software Engineering, A Practitioner's Approach, McGraw-Hill, 2010.

2. Abelson, Linda A., Suellen Eslinger, Marvin C. Gechman, and Carol H. Ledoux. *Software Measurement Standard for Space Systems*. TOR-2009(8506)-6, The Aerospace Corporation, El Segundo, CA. 2011.

3. McGarry, John, David Card, Cheryl Jones, Beth Layman, Elizabeth Clark, Joseph Dean, Fred Hall. *Practical Software Measurement: Objective Information for Decision Makers*. Addison Wesley, October 2001.

4. Fenton, Norman E. and Shari Lawrence Pfleger. *Software Metrics: A Rigorous and Practical Approach*, 2nd ed. London, UK; International Thompson Computer Press. 1997.

5. Gaffney, et al. *The Software Measurement Guidebook*. International Thompson Computer Products, 1995.

6. McConnell, Steve. *Software Estimation: Demystifying the Black Art*. Redmond, Washington: Microsoft Press. 2006.

7. Office of the Deputy Under Secretary of Defense for Acquisition and Technology, Systems and Software Engineering, Enterprise Development. Systems Engineering Plan Preparation Guide, Version 2.01. Washington, DC: ODUSD(A&T)SSE/ED, 2008.

8. SMC System Engineering Primer and Handbook, Vol. 1, 4th Edition, March 11, 2013.

9. Adams, R. J., S. Eslinger, K. L. Owens, J Tagami. *Software Development Standard for Mission Critical Systems*. TR-RS-2015-00012. The Aerospace Corporation, El Segundo, CA. March 2014.

10. Basili, V., and D. M. Weiss. "A Methodology of Collecting Valid Software Engineering Data." *IEEE Transactions on Software Engineering*, pages 728-738. November 1984.

11. Basili, V., and Gianluigi Caldiera "Goal Question Metric Paradigm," *Encyclopedia of Software Engineering* – 2 Volume Set, 1994.

12. ISO/IEC 15939. International Organization for Standardization/ International Electrotechnical Commission (ISO/IEC), Systems and

Software Engineering – Measurement Process, ISO/IEC 15939:2007, 1 August 2007.

13. Houston, D., N Kern, K. Sharp, B. Troup, C. Wang, Recommended Software Measures for MILSATCOM Sustainment Programs, TOR-2013-00750. The Aerospace Corporation, El Segundo, CA.

14. Roedler, G, Donna H. Rhodes, Howard Schimmoller, Cheryl Jones, Systems Engineering Leading Indicators Guide, INCOSE-TP-2005-001-03, January 2010

15. CMMI Product Team. CMMI for Development, Version 1.3 (CMU/SEI-2010-TR-033). Software Engineering Institute, Carnegie Mellon University, 2010. CMMI® is registered in the U. S. Patent and Trademark Office by Carnegie Mellon University.

16. Adams, Richard J., Suellen Eslinger, Karen L. Owens, Joanne M. Tagaml. *Software Development.* TR-RS-2015-00012 (SMC-S-012), The Aerospace Corporation, El Segundo, CA. 2014.

27.10 Bibliography

DI-IPSC-81427A, Software Development Plan (SDP), 10 January 2000.

DI-SESS-81785, Systems Engineering Management Plan (SEMP), 14 October 2009.

DI-MGMT-80368, Status Report, 30 October 2006.

27.11 Acronyms

AO	availability
BMI	backlog management index
CCB	configuration control board
CCP	contract change proposal
CDA	critical design audit
CDR	critical design review
CDRL	contract data requirements list
CMMI	capability maturity model integration
CPU	central processing unit
CTM	critical technical parameters
DID	data item descriptions
ECP	engineering change proposal
FCA	functional configuration audit

FQR	formal qualification review
FRR	flight readiness review
GQM	goal question metric
IBR	integrated baseline review
IEC	International Electrotechnical Commission
INCOSE	International Council on Systems Engineering
IRRT	independent readiness review team
ISO	International Standards Organization
KDP	key data processor
KPP	key performance parameters
KSA	key system attributes
LRR	launch readiness review
MAG	*Mission Assurance Guide*
MAP	measurement and analysis plan
MOE	measures of effectiveness
MOP	measures of performance
MOS	measures of suitability
MRR	mission readiness review
PCA	physical configuration audit
PDA	parallel data adapter
PDR	preliminary data review
PRR	production readiness review
PSM	practical software and systems management
RFP	request for proposal
SDP	software development plan
SDR	system design review
SE	systems engineering
SEMP	system engineering management plan
SEP	system engineering plan
SLOC	source lines of code
SMART	software measurement analysis and repository toolset
SME	subject matter expert
SMP	software measurement plan
SMP	software measurement plan
SMR	software measurement report
SOW	statement of work
SPO	system program offices
SRR	system requirements review
SVR	system verification review
TBX	to be (variable)
TiP	time in process
TPM	technical performance measures
UOM	unit of measurement
WBS	work breakdown structure

Chapter 28
Ground Quality Assurance

Arthur L. McClellan
Systems and Operations Assurance Department
Mission Assurance Subdivision
Dana J. Speece
Corporate Quality Management Office
Corporate Chief Engineer's Office

28.1 Introduction

Ground quality assurance (GQA) is the engineering and management specialty discipline that implements the planned and systematic activities in a ground quality system so that ground quality requirements for a product or service will be fulfilled. One of the primary goals of GQA is to ensure availability and readiness of ground segment elements to support deliverable products and services.

GQA begins in the early program phase with the development of requirements and establishment of the systems design concept. It continues through the program lifecycle to the operations and sustainment phase. A quality assurance program provides an organizational framework and identifies which areas are most conducive to ensuring product availability and readiness. When well defined and implemented, a ground quality assurance program ensures that all ground quality requirements are verified through assessment of analyses, operations, processes, procedures, testing, and inspection.

28.2 Definitions

Product or service quality The degree to which the product or service attributes, such as capability, performance, and reliability, meet the needs of the customer or mission, as specified through the requirements definition and allocation process.

Quality assurance (QA) The engineering and management specialty discipline that defines the standards to be followed to meet the product or service customer requirements. It implements planned and systematic activities in a quality system to accomplish this. QA consists of preventive actions which are focused on processes and procedures and are ideally done prior to the delivery of the product or service.

Quality Program Plan (QPP) The contractor shall describe in a QPP the approach for managing and implementing the quality requirements of the

program. The QPP often addresses design reviews, fabrication, audits (internal and external), failure reporting, and the corrective action system. It usually will address the approach for the control of parts, materials, processes, reliability, software, supplier control, quality, test, and delivery.

Software Quality Plan For deliverable software the contractor's approach to the program's software quality requirements shall comply with a customer-approved software quality plan. For contractor-developed nondeliverable software used to manufacture or test deliverable hardware or software, the contractor shall implement a disciplined management system for the validation and maintenance of such nondeliverable software. The software quality program approach shall be managed as a part of, and be consistent with, the general requirements for the overall quality program plan.

Special Test Equipment Non-deliverable special test equipment (STE) acquisition, design, fabrication, or modification may require special-purpose testing. STE hardware, such as electrical ground support equipment (EGSE) and mechanical ground support equipment (MGSE), and software undergo design review, inspection, and acceptance test prior to interfacing with flight hardware.

Control of Nonconforming Product Quality assurance identifies, documents, segregates, evaluates, and dispositions all nonconforming products that do not conform to drawings, specifications, or other engineering and technical acceptance criteria, to prevent their unintended use. The reporting system provides real-time status and control of nonconforming products.

28.3 Broad Description of Ground Quality Assurance

GQA ensures that the elements of the ground segment, which include the ground station and terminals, mobile ground equipment, launch systems integration, network and range interfaces, and the mission control center, are verified, assessed, and are ready and available to support their intended function.

GQA tasks are typically the responsibility of, and completed by, government contractor's personnel. Contractor quality engineering defines and supports the implementation of the program or project QA activities as defined by the contract and the quality program plan. The government program office team establishes QA requirements for each program via contract and also verifies conformance to those requirements. The contractor also flows QA requirements to subcontractors and suppliers to assure the successful execution of all QA activities for ground segment programs.

GQA employs a consistent methodology in the verification and validation (V&V) of each ground segment element. Figure 28-1 identifies some of the major tasks that GQA will focus on. Within this framework, GQA identifies the

requirements for V&V, then plans the what, how, when, and where to do their tasks. Key to the GQA function and task is their interface with the program office. Often that customer is focused on their hardware performance or final product, and not the infrastructure of the ground segment. The timeliness of the program office interface with GQA is critical to assure the planning and the required time is allocated for GQA to effectively do their tasks of V&V.

Following are brief descriptions of specific GQA activities required for each ground segment element.

28.3.1 Ground Station Terminals

There are several functions of a ground station: command and control, data handling, and routing. GQA ensures the equipment is configured for the mission through software verification and configuration management. As part of the availability of the ground station and terminals there are multiple terminals for command and control and payload operation within the ground station. GQA must ensure all hardware constituents are configured the same, operated the same, access correct command database files from the servers, and are interchangeable to serve in a backup scenario. GQA also ensures the internal interfaces are valid, and the terminals can retrieve mission data for playback as part of a mission anomaly. Additionally, GQA assures mission data security protocol is verified and maintained and that the crew exercise simulations and operator training meet mission requirements. Facility air handling and backup power are key items for reliable station operation.

28.3.2 Mobile Ground Element

GQA ensures that the same areas assessed for the ground station, are again verified for the mobile ground element. Configuration, physical security, facilities, operator training, secure data handling, and server controls are just a few of the features that GQA will V&V. Additional GQA assessment areas are transportability of the mobile station, bringing the station online after transportation, external power availability, how the station is certified for operation, unique data handling requirements, link analyses for specific locations, specific facility needs including physical security, and crew support.

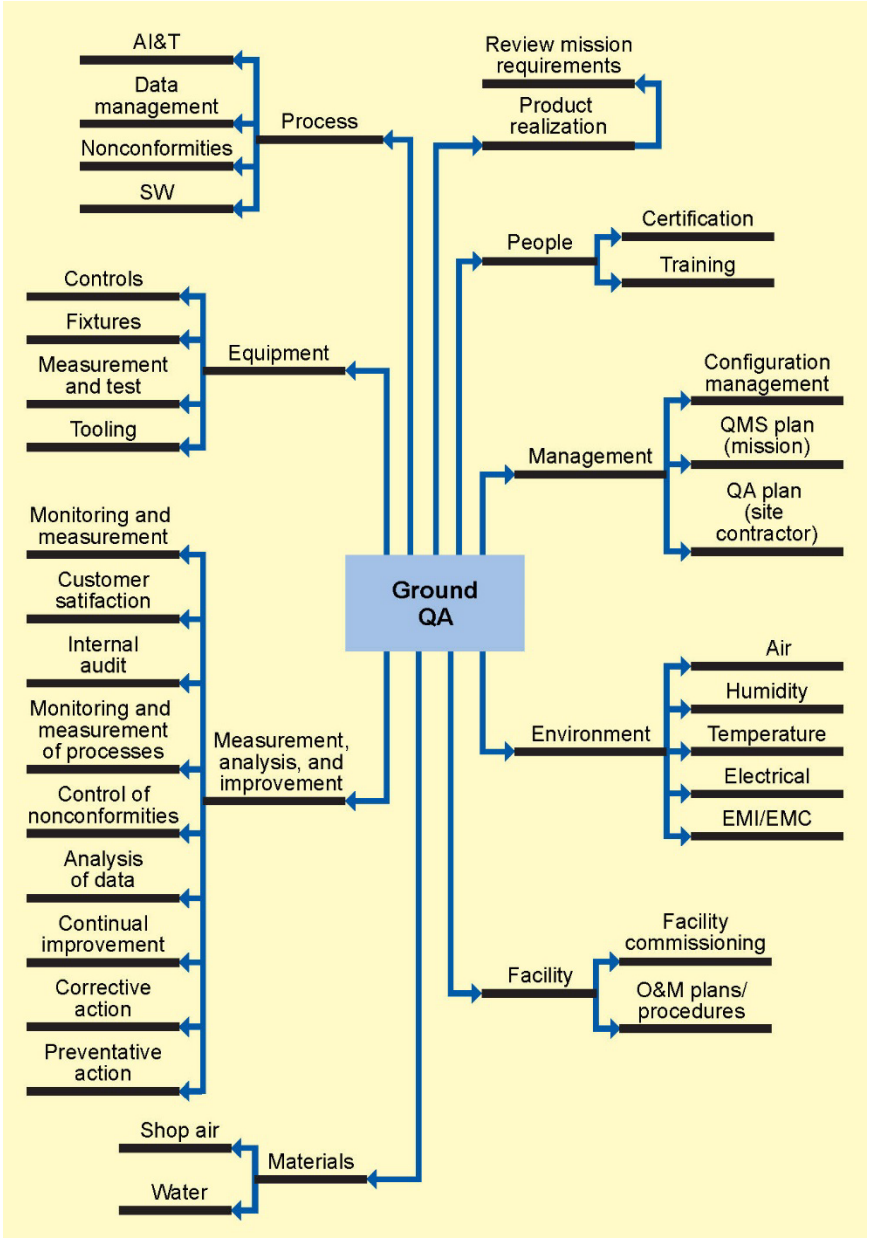


Figure 28-1. Ground QA relationship to QA activities.

28.3.3 Launch System Integration

GQA coordinates with the program office to assure the requirements for launch systems integration have been properly planned and scheduled. The ground segment must be ready to support during all facets of the launch vehicle and space vehicle integration. Testing, crew exercises, pad operations, facility equipment, MGSE, EGSE, special operations, and safety processes are all areas in which GQA ensures readiness. There are several network interfaces, as identified by the program office, that GQA assures are ready and available.

28.3.4 Network and Range Interfaces

While the space vehicle (SV) is preparing for integration and launch, GQA must ensure SV communications with all other segments. Readiness and availability of the network and range interfaces from the SV to the operations centers, both fixed and mobile sites (including ground terminals), is mandatory to launch.

GQA will verify requirements are in place through a review of the segment ICDs, space to ground, ground entry point to operations centers, and operations centers to the mission data processing center. GQA assures the necessary control elements and data products are in place for the measurement and analysis of the numerous verifications, including tests and inspections, as they are planned and executed. The plans, training, and certifications are verified to ensure the appropriate personnel are in the right place at the right time. Control of the terminals, software, and configurations are crucial because there are always changes to facilities, equipment, and the environment.

28.3.5 Mission Control Center

Without the mission control center (MCC) there is no mission. The MCC primary elements are mission management, space and ground asset command and control, mission data processing and distribution, and other additional functions which provide support and operation of the mission satellite. Mission management provides the planning, tasking, scheduling, and performance assessment of satellite operations and assets needed to maintain satellite operations. This is where GQA can see the daily plan of operations and other significant event status and identify needed quality tasks. Space and ground asset command and control has the primary function of satellite command and control, ground asset command and control, anomaly response, sustaining engineering, and space to ground communications. GQA must ensure the availability of the ground assets to support these functions.

Mission data processing and distribution is the reason for the mission, and as such has special requirements for GQA to ensure compliance to those requirements. Mission data capture, handling, encryption/decryption, processing, data security,

and distribution all have specific mission requirements. Months ahead of launch the mission requirements are identified and planned for execution. This is where GQA must identify in each area what, how, where, and when they will verify and assess compliance to those requirements.

Additional functions within the MCC include archive and retrieval, simulation, test, and training. These activities support mission-unique requirements from the MCC to the operators within the ground stations, which includes the training, testing to assure compliance and function, and the overall mission data archive and retrieval. GQA must ensure that the functions identified are properly assessed.

28.4 Technical Considerations

The relationship between the traditional QA activity areas identified in ISO 9001 [1] and the related tasks for GQA are graphically displayed in Figure 28-1. QA performs planned and systemic activities to assure that the infrastructure is controlled to meet requirements. Table 28-1 lists each activity area and provides summary descriptions of related QA tasks. The activities from Figure 28-1 and Table 28-1 are illustrated in Figure 28-2. The figure emphasizes that requirements need to be considered at the beginning of a program and that there are recurring facility activities throughout the program life cycle.

Table 28-1. Ground QA Activity Areas and Tasks

Activity Area	QA Tasks
Product realization	Review requirements for the mission to determine what needs to be tailored in the other activities to meet customer needs. For example, a review of requirements may result in the need to provide certain training for operating personnel. There are industry standards that set the requirements and/or provide guidance for the construction, maintenance, and operating plans. However, the various government agencies often have their own standards and guidelines for establishing and implementing these plans. The customer, along with the appropriate subject matter experts, should verify which standards apply.
People	Assure that personnel have the appropriate certification and/or training for tasks that they are required to perform.

Activity Area	QA Tasks
Management	Develop configuration management plans or procedures to establish, record, and update changes for the baseline functional and physical attributes for the facility, and processes, software, tools, and equipment, etc., that impact flight hardware and/or the mission. Develop quality management plans for the facility in accordance with an industry standard such as ISO 9001. As required, develop a quality program plan to implement customer specific requirements.
Environment	Assure that plans are implemented to control the environment (air, humidity, temperature, electrical, EMI/EMC) in order to prevent damage (e.g., contamination) to the facility, equipment, personnel, and flight hardware.
Facility	Commissioning is an all-inclusive process for working with project teams to document the planning, delivery, verification; and to manage risk throughout the lifecycle. It ensures facility quality using design review and verifications, usually through testing, analysis, and/or inspections. Commissioning establishes the benchmark of performance and helps develop the operations and maintenance (O&M) documentation that form the foundation for the facility's O&M program and the building's lifecycle. O&M plans should be established that outline guidance for managing and executing day-to-day O&M activities. There are five areas that encompass an effective O&M program structure: • operations • maintenance • engineering support • training • administration and management Comprehensive O&M manuals should be developed and maintained which include: • system-level O&M information with physical and functional descriptions of the systems • troubleshooting • preventive maintenance procedures and schedules (intervals) • corrective maintenance requirements • real property inventory and parts and spares lists • as-built drawings/documentation and O&M significant diagrams

Activity Area	QA Tasks
Materials	Assure that plans are implemented to control materials that contact flight hardware, or operational equipment, to prevent degradation or damage.
Measurement, analysis, and improvement	This tasks entails establishing controls and metrics to obtain performance feedback to drive continuous improvement in the facility and its operations. Controls are typically applied in the following areas as indicated in industry quality standards, e.g., ISO 9001: • monitoring and measurement (e.g., equipment) • customer satisfaction • internal audits • monitoring and measurement of processes (e.g., assembly, integration, and test) • control of nonconformities (e.g., facility outages, nonconformities during payload processing) • analysis of data (e.g., cycle time for payload processing) • continual improvement • corrective action As part of yearly strategic planning and long-term sustainment planning, facility audit plans should also be developed that outline the in-depth review and observation of the facilities and their critical systems. The audit plan should, at a minimum: • evaluate the condition of each facility and its systems and assess the ability and capability to support missions • identify any deficiencies and/or less than optimum conditions for each building system • provide periodic inspections of the facility and systems to verify whether corrective actions have been implemented and how they are performing (have issue been mitigated and expectations/ requirements) and if expectations are met
Equipment	Develop plans to assure that configuration for tooling, measurement and test equipment, fixtures, and controls (e.g., power) is known at all times in order to prevent usage of nonconforming items – e.g., uncalibrated test equipment.

Activity Area	QA Tasks
Process	This task entails establishing measures to control variation in items, or areas, to prevent nonconformities in the facilities operation and product (flight hardware, data, etc.). Typically controlled areas are AI&T, data management, nonconformance processing, and software development and usage.

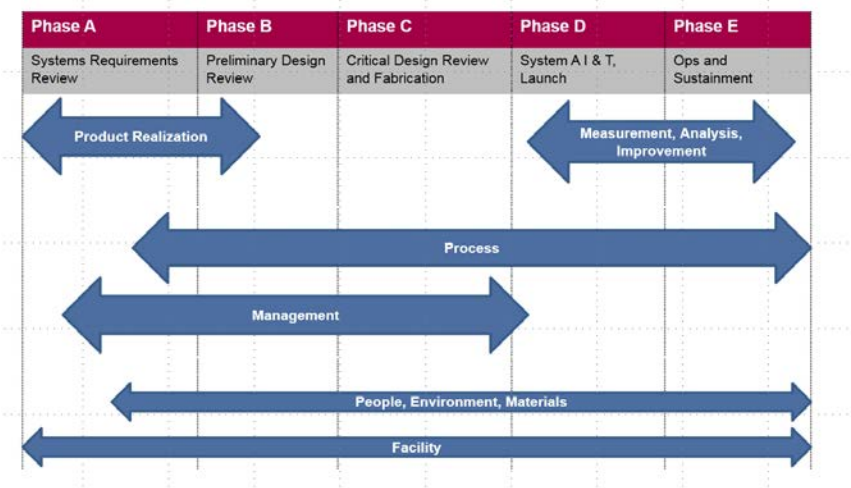


Figure 28-2. Ground QA activities across program lifecycle.

28.5 Programmatic Considerations

Table 28-2 lists some common pitfalls due to neglecting quality assurance activities.

Table 28-2. Pitfalls of Neglecting QA Activities

Activity	Pitfalls
Product realization	Inability to process mission data due to inadequate power for equipment such as computers, antennas, etc. Outages due to failure to design for the ability to perform preventive maintenance with a plan that doesn't require a shutdown.
Measurement analysis and improvement	Recurring outages due to inadequate root cause determination and the inability to implement appropriate corrective action Failure to collect metrics for nonconformities that:

Activity	Pitfalls
	• identify the root cause • document corrective action • indicate if a problem is recurring • categorize data at a high enough level to reveal trends (.e.g., binning at a fine detail that results in a flat histogram)
Process	Special cause variation in process and unpredictable output resulting in nonconformities such as damaged flight hardware, unexpected software behavior, loss and/or corruption of mission data
People	Nonconformities such as outages due to operation of support equipment (e.g., power systems) by untrained or uncertified personnel
Facility	Outages due to failure to perform preventive maintenance per an established O&M plan

28.6 Reference

1. *Quality Management Systems – Requirements*, Organization for International Standardization. ISO 9001:2015. September 15, 2015.

28.7 Bibliography

Specification and Standards

Quality Management Systems – Fundamentals and Vocabulary, Organization for International Standardization. ISO 9000:2015. September 15, 2015.

Quality Management Systems – Requirements for Aviation, Space and Defense Organizations, SAI International. SAE AS9100. January 15, 2009.

Richter, Eric S. *Quality Space and Launch Requirements Addendum to AS9100C*, TR-RS-2015-00003, The Aerospace Corporation, El Segundo, CA. March 5, 2015

Eslinger, Suellen, Karen L. Owens, and Joanne M. Tagami. *Software Development Standard for Mission Critical Systems*, TR-RS-2015-00012, The Aerospace Corporation, El Segundo, CA. March 17, 2014.

Guaro, Sergio B., Gail A. Johnson Roth, and William F. Tosney. *Mission Assurance Guide*, TOR-2007(8546)-6018, Rev B. The Aerospace Corporation, El Segundo, CA. June 1, 2012.

28.8 Acronyms

AI&T	assembly, integration, and test
EGSE	electrical ground support equipment
EMC	electromagnetic compatibility
EMI	electromagnetic interference
GQA	ground quality assurance
ICD	interface control document
ISO	International Standards Organization
MCC	mission control center
MGSE	mechanical ground support equipment
O&M	operations and maintenance
QA	quality assurance
QPP	quality program plan
STE	special test equipment
V&V	verification and validation

Chapter 29
Ground Segment Readiness Reviews

Suellen Eslinger
Software Engineering Subdivision
Computers and Software Division

29.1 Introduction

The readiness review process is a series of systematic and coordinated review and assessment activities to assure that subsystems and systems being acquired successfully complete specified readiness milestones. The readiness review defines the primary framework for the planning, execution, and closure of the system acquisition process. Readiness reviews occur throughout the ground segment development life cycle.

29.2 Definitions

Architecture The fundamental organization of the system or software embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution [1].

Build A version of software that meets a specified subset of the requirements that the completed software will meet [1].

Functional requirement A requirement that defines a specific behavior or function of the system or software. In general, functional requirements define what a system is supposed to do, while non-functional requirements define how a system is supposed to be [1].

Hardware item An aggregation of hardware that satisfies an end use function and is designated for specification, interfacing, qualification testing, configuration management, or other purposes [1].

Mission assurance The disciplined application of proven scientific, engineering, quality, and program management principles toward the goal of achieving mission success [2].

Mission success The achievement by an acquired system (or system of systems) to singularly or in combination meet not only specified performance requirements but also the expectations of the users and operators in terms of safety, operability, suitability and supportability [3].

Non-functional requirement A system or software requirement that specifies criteria that can be used to judge the operation of the system or software, rather than specific behaviors. Contrast with functional requirements that define specific behavior or functions. Performance requirements, including response times, throughput, hard time deadlines, and accuracies, are considered non-functional requirements. Other examples of non-functional requirements are specialty engineering requirements (e.g., security, safety, reliability, maintainability, availability, and human systems integration), computer resource margin requirements, and scalability and extensibility requirements. Adapted from [1].

Performance requirement See non-functional requirement.

Program management baseline A time-phased budget plan for accomplishing work against which contract performance is measured [4].

Software item An aggregation of software that satisfies an end use function and is designated for specification, interfacing, qualification testing, configuration management, and other purposes [1] .

Technical review A series of systems engineering activities conducted at a logical transition point in a system life cycle, by which the progress of a program is assessed relative to its technical requirements using a mutually agreed-upon set of criteria [5].

29.3 Objectives

The objective of these reviews is to determine the readiness of the acquisition to proceed to the next phase of the acquisition lifecycle using defined entrance and exit criteria.

29.4 Readiness Review Process (Common Activities)

Each of the types of readiness reviews described in this chapter contains three steps: review planning and preparation, review execution, and review closure. While the details of performing readiness reviews differ for each type, there are activities that are common for these three steps across all types of readiness reviews (Table 29-1).

Table 29-1. Readiness Review Activities

Readiness Review Step	Principal Common Activities
Review Planning and Preparation	a. Definition of review processes to be followed, agreed to by the review team and contractor team b. Identification of evaluation criteria or review exit criteria to be used for the review, agreed to by the review team and contractor team c. Development of review schedule and agenda, agreed to by review team and contractor team d. Identification of methods and tools to be used to document action items to be addressed, issues to be resolved, and discrepancies/deficiencies to be corrected e. Preparation of contractor documentation to be reviewed, sufficiently in advance of the review date to allow for thorough review by the review team f. Preparation of contractor presentations to be given to the review team during the review g. Preparation of demonstrations to be given to the review team during the review h. Identification of personnel involved in the review (both review team and contractor team) i. Establishment of an organized repository of all program documentation needed for the review, both those documents under review and those needed for reference, under appropriate security controls but easily accessible by the review team j. Establishment of review logistics (e.g., presentation rooms, documentation review rooms, demonstration facilities, test facilities, as needed)
Review Execution	a. Documentation review by the review team b. Presentation of material by the contractor team to the review team, and review of the material presented by the review team c. Demonstrations by the contractor team to the review team, and review of the demonstrations by the review team d. Review team assessment of the information provided by the contractor team against the pre-defined evaluation criteria (or review exit criteria) e. Identification and documentation of action items to be addressed, issues to be resolved, and discrepancies/deficiencies to be corrected (by both the review team and contractor team)

Readiness Review Step	Principal Common Activities
Review Closure	a. Determination of overall assessment by the review team b. Delivery of overall assessment by the review team to the contractor team, or preparation of an assessment summary briefing by the review team and delivery of that briefing to the contractor team c. Assignment of responsible party(ies) and due dates for all action items d. Assignment of responsible party(ies) and due dates for resolution of all issues e. Assignment of responsible party(ies) and due dates for evaluation of all discrepancies/deficiencies and proposal for correction f. Completion of action item responses, issues resolutions, and discrepancy/deficiency correction by the contractor (i.e., the organization on whom the readiness review was performed, usually the contractor) g. Follow-up of action item responses, issue resolutions, and discrepancy/deficiency corrections by the sponsor (i.e., the organization that required the readiness review, usually the acquirer) h. Documentation of readiness review closeout by the sponsor after all action items, issues, and discrepancies/deficiencies are satisfactorily handled

29.4.1 Key Lessons Learned

The following general lessons learned apply to all readiness reviews addressed in this chapter:

- To ensure that the readiness reviews can be conducted without additional cost to the acquirer, the contract should include requirements for the contractor to prepare for and/or perform the reviews, as appropriate. The readiness reviews can be required by the government statement of work, mandated by a compliance standard, or included as a special provision (Section H). See Abelson et al. [6] for additional information.
- The entrance criteria, exit criteria, and review agenda should always be agreed to by all parties in advance of the review.
- The contractor should maintain documentation on the action items, issues, and problems identified by the readiness reviews and should ensure there is a closed-loop process for resolving these items. The

acquirer should carefully follow up on all action items, issues, and problems identified by the readiness reviews to ensure that they are satisfactorily resolved in a timely manner.

- Satisfactory accomplishment of the readiness reviews and satisfactory, timely resolution of all action items, issues, and problems resulting from the reviews should be included as incentives on the contract.

Lessons learned for each specific type of readiness review will be presented in the following sections.

29.5 Key Readiness Reviews

This section describes the following types of readiness reviews: program management reviews, major technical reviews, software-specific joint technical reviews, deployment and operations reviews, and independent assessments. This chapter does not address deactivation and disposal reviews.

29.5.1 Program Management Reviews (PMRs)

PMRs address such issues as schedules, budgets and costs, earned value, risks, and other management-related topics. Some examples of PMRs include weekly or monthly management reviews between the contractor and acquirer for specific hardware or software items, monthly reviews between contractor program management personnel and acquirer program management personnel, monthly reviews between the acquirer program manager and higher level acquisition managers, and formal acquisition reviews (such as milestone reviews) with acquisition authorities.

The only type of PMR to be discussed in this section will be the integrated baseline review (IBR), which focuses on the technical and management work to be performed in a ground segment development.

29.5.1.1 Integrated Baseline Review (IBR)

The IBR is a program management review that is important to the successful systems engineering and development of any ground segment. The IBR is a joint assessment of the program management baseline (PMB) by the contractor and acquirer to ensure the mutual understanding of the scope of the work to be performed; the management control processes to be used; the risks associated with cost, schedules, and resources; and corrective actions where needed. IBRs are scheduled within 6 months of contract award and whenever major program changes occur thereafter.

During the IBR, the contractor's work breakdown structure (WBS) is reviewed to ensure it contains all of the work on the program. A ground segment WBS should include, as a minimum, separate top-level divisions for systems engineering, development, integration and test, transition to operations, documentation (called "data"), and management [7]. Because most ground segments are developed in increments using an iterative type of lifecycle model, the WBS must provide for separate accounting of each ground segment increment if there is more than one. Next, the WBS must have separate items for each architectural component, broken down hierarchically into the sub-components that constitute the program. Finally, the WBS must account for the types of work being performed for each lowest level sub-component (e.g., architecture, detailed design, implementation/fabrication or procurement, integration and test, and verification). From a ground segment systems engineering perspective, the contractor's WBS must be reviewed to ensure it properly represents the ground segment increments, the contractor's ground segment architecture, and the technical work to be performed.

Other important items to be reviewed from a systems engineering perspective are the contractor's effort and schedule and the method the contractor plans to use for earned-value for each of the lowest level WBS items (called "control accounts"). The effort and schedule must be sufficient to accomplish the amount of work in each of the control accounts, and the earned-value method must be consistent with the work to be performed. In addition, reviewing the networked schedule built-up from the lower level schedules with dependencies is essential for ensuring the feasibility of the contractor's plans for developing the ground segment.

Table 29-2. IBR Objectives

Principal Objectives	Major Review Products
a. "The technical scope of work is fully included and is consistent with authorizing documents" b. "Key project schedule milestones are identified" c. "Supporting schedules reflect a logical flow to accomplish the work" d. "Resources (budgets, facilities, personnel, skills, etc.) are adequate and available for the assigned tasks" e. "Tasks are planned and can be measured objectively, relative to the technical progress" f. "Underlying PMB rationales are reasonable" g. "Managers have appropriately implemented required management processes"	<u>Government Products:</u> a. Contract b. Contract modifications <u>Contractor Products:</u> a. Contractor statement of work b. Contractor WBS c. Scope, budgets, schedules, earned value methods for each control account d. Integrated master plan, integrated master schedule e. Detailed networked schedules from control account schedules up to the overall project schedule

Principal Objectives	Major Review Products
h. The risk in the ground segment PMB is at an acceptable level to proceed with further contract development activities i. The management reserve is consistent with current project risk not accounted for by the PMB [8]	f. Contractor management control processes

29.5.1.1.1 Execution by Acquisition Phase (IBR)

The IBR occurs shortly after contract award, usually within six months of receiving authority to proceed (ATP). IBR repeats throughout the contract if major contract modifications are made.

Development of a new ground segment may require proceeding through several acquisition phases. Each acquisition phase may involve a competitive source selection and a new contract. In this case, each new contract will have at least one IBR. Smaller ground segments, or upgrades of existing ground segments, may require only one development contract, and therefore one IBR, unless major contract modifications are made.

29.5.1.1.2 IBR Lessons Learned

The following lessons learned apply to the IBR:

- While the IBR is concerned with cost and schedule, it also requires a deep technical knowledge on the part of the acquirer and contractor review team. For each cost account, the preparers of the estimates and the reviewers must understand the extent of the technical work to be done, how the cost and schedule for that work are estimated, and how progress for that work will be measured (i.e., the technique for earning value). Therefore, the IBR team should never be staffed solely with cost and schedule personnel who do not understand the technical work to be performed on the contract.

- Training of the IBR team is essential, both for the contractor team and the acquisition team. No one should be on the IBR team who does not understand cost and schedule estimation of the technical work, earned value, and the contractual requirements.

- Because technical experience is needed to understand the extent of the technical work and how to estimate it, the review team must include experienced technical personnel for each cost account. Junior personnel

should never be the sole people preparing or reviewing any cost account. This applies to both contractor and acquirer personnel.

- Because most ground segments are developed in increments, multiple IBRs should be performed as more information is known about the contents, cost, and schedule of each increment. At least one IBR per ground segment increment should be held, close to the beginning of development of the increment.

Guidance on the IBR process, including additional lessons learned, can be found in *The Program Manager's Guide to the Integrated Baseline Review Process* [9].

29.5.2 Major Technical Reviews

Technical reviews are defined as "a series of systems engineering activities conducted at logical transition points in a system life cycle, by which the progress of a program is assessed relative to its technical requirements using a mutually agreed-upon set of criteria" [5]. A major technical review is a technical review that is a significant event in the system development lifecycle, used to determine readiness of the developer to proceed with downstream engineering activities and to provide the acquirer with sufficient understanding of the developer's technical progress and residual technical risks to make program decisions.

The major technical reviews that apply to a typical ground segment development include system requirements review (SRR), system design review (SDR)/ system functional review (SFR), software requirements and architecture review (SAR), preliminary design review (PDR), critical design review (CDR), and test readiness review (TRR). While other major technical reviews are possible [such as alternative system review (ASR) and production readiness review (PRR)], this chapter addresses only these six major technical reviews. Note that the second review in the above list is known by either name, system functional review, or system design review, depending upon the acquisition organization's culture and regulations.

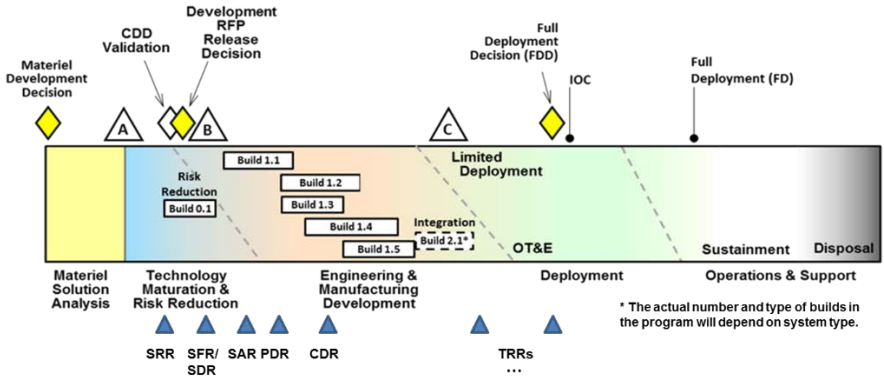
Additional information on major technical reviews, including entry and exit criteria, products to be reviewed, and detailed acceptability criteria, can be found in works from the IEEE and Peresztegy et al. [5, 10-11].

29.5.2.1 Execution by Acquisition Phase (Major Technical Reviews)

The six major technical reviews described in section 36.5.2.2 occur in an ordered sequence across the development lifecycle. In this section, the placement of the

major technical reviews is addressed for the two DOD lifecycles most applicable to ground segments.

Figure 29-1 shows an example of where in the lifecycle the major technical reviews are usually located for a defense-unique software intensive program. Because each program is different, the placement of the major technical reviews may vary according to the acquisition strategy of a particular ground segment. The major technical reviews, however, should be held in the sequence shown in Figure 29-1. If there is a new contract for the engineering and manufacturing development (EMD) phase, the SAR and PDR may be held again at the beginning of the EMD contract to address any requirements changes that have occurred since the technology maturation and risk reduction phase.



CDD = capability development document; RFP = request for proposal; IOC = initial operational capability;
OT&E = operational test and evaluation

Figure 29-1. Defense unique software intensive program with major technical reviews [2].

The other DOD lifecycle model most applicable to ground segments is the incrementally fielded software intensive program. In this case, the lifecycle repeats for each ground segment increment, and the major technical reviews would be repeated for each increment in the sequence and position shown in Figure 29-1 for the single-increment, defense-unique software intensive program.

29.5.2.2 Types of Major Technical Reviews

29.5.2.2.1 System/Segment Requirements Review (SRR)

The ground SRR is a multidisciplinary review to ensure that the ground segment requirements are sufficiently well defined to proceed with further requirements

and design activities. Following a successful ground SRR, the developer's ground segment requirements specification should fully address all ground segment contractual requirements.

In large systems, the ground segment may be only one segment of the system. In this case, there is usually an SRR that applies to the entire system and addresses the System Requirements Specification. The system requirements are then allocated to the segments, and segment level requirements are elaborated and reviewed at the segment SRRs. Following a successful Ground SRR in this context, the developer's Ground Segment Requirements Specification should fully address all system requirements allocated to the ground segment.

The objectives of the Ground SRR are shown in Table 29-3.

Table 29-3. SRR Objectives

Principal Objectives	Major Review Products
a. The ground segment requirements, including interface requirements, are sufficiently well defined and documented to proceed with further requirements analysis, decomposition, allocation, and elaboration and with further ground segment design b. The ground segment requirements address all system/contractual requirements allocated to the ground segment, including all functional and non-functional requirements, as demonstrated by the completed and correct bidirectional traceability between the ground segment requirements in the Ground Segment Specification and the system and/or contractual requirements allocated to the ground segment c. A candidate ground segment architecture is presented that will meet the ground segment requirements and satisfy the user's concept of operations d. The ground segment requirements are feasible given available technologies that are sufficiently mature, or will be sufficiently mature by PDR	• Initial capabilities document (ICD) • Capability development document (CDD) • Capability production document (CPD) • Concept of operations (CONOPS) • Technical requirements document (TRD) [or system requirements document (SRF)] • Analysis of alternatives (AoA) • Preferred system architecture • Government reference architecture (GRA) System threat assessment report (STAR) • Risk assessment • Technology development strategy (TDS) • Critical technology elements (CTEs) • Technology readiness assessment (TRA) • Test and evaluation strategy (TES) • Test and evaluation master plan (TEMP) • Acquisition strategy plan (ASP)

Principal Objectives	Major Review Products
e. The ground segment development plans are feasible and effective for building the ground segment under contract f. The ground segment test planning for verification of the ground segment requirements is sufficiently well defined to guide the later development of ground segment test procedures and provides traceability of all ground segment requirements, including functional, non-functional, and interface requirements, to well defined tests g. The estimated ground segment cost and schedule are within program constraints h. The ground segment development risk is at an acceptable level to proceed with further requirements and design activities	• Intellectual property (IP) plan (documents as part of the ASP) • Systems engineering plan (SEP) • Lifecycle sustainment plan (LCSP) • Information support plan (ISP) • Program protection plan (PPP) cybersecurity plan (documented as an appendix to the PPP) Software assurance plan (documented as an appendix to the PPP) • Programmatic environment, safety, and occupational health evaluation (PESHE) • Software acquisition management plan (SWAMP) Clinger Cohen Act compliance assessment • Cost analysis requirements document (CARD) • Requests for proposal (RFPs) • Acquisition milestone documentation or acquisition decision point documentation <u>Contractor Products:</u> a. System Requirements Specification b. Ground Segment Requirements Specification c. Draft Ground Segment Architecture d. Bidirectional traceability between the contractual requirements, system requirements, and segment requirements e. System Test Plan and Ground Segment Test Plan f. Initial engineering analyses, simulations, models, prototypes g. Initial technology readiness evaluation h. Ground segment development plans (e.g., ground segment systems engineering, security, reliability, safety, human systems integration, risk management, sustainment, and transition plans) i. Ground segment cost and schedule

Principal Objectives	Major Review Products
	j. Ground segment risk list and handling plans (including mitigation plans) for risks ranked medium to high

29.5.2.2.2 System/Segment Design Review (SDR) (a.k.a. System/Segment Functional Review [SFR])

The Ground SDR/SFR is a multidisciplinary review to ensure that the ground segment can proceed to preliminary design and that all functional and non-functional requirements, including performance and specialty engineering requirements, are well defined and consistent with the ground segment contractual requirements, cost, and schedule. Following a successful Ground SDR/SFR, the Ground Segment Requirements Specification should be ready to be baselined.

The ground segment architecture is defined and reviewed at the Ground SDR/SFR. For large ground segments, the architecture usually divides the segment into major elements or subsystems. In this case, the ground segment requirements are allocated to the elements/subsystems and the element/subsystem level requirements are elaborated and reviewed at the SDR/SFR. Following a successful SDR/SFR in this context, the element/subsystem requirements should fully address their allocated ground segment requirements. Usually there is only one ground SDR/SFR, even in large ground segments where there are multiple ground elements/subsystems.

Table 29-4. SDR/SFR Objectives

Principal Objectives	Major Review Products
a. The ground segment architecture is sufficiently well defined and documented to proceed with further ground segment requirements and design activities b. The ground segment architecture defines the ground segment elements/ subsystems and their constituent hardware and software items as well as the interfaces among the elements/subsystems and among the hardware and software items c. The ground segment requirements are consistent with the ground	• Initial capabilities document (ICD) • Capability development document (CDD) • Capability production document (CPD) • Concept of operations (CONOPS) • Technical requirements document (TRD) [or system requirements document (SRF)] • Analysis of alternatives (AoA) • Preferred system architecture • Government reference architecture (GRA)

Principal Objectives	Major Review Products
segment architecture and are ready to be baselined i. The ground element/subsystem requirements, including interface requirements, are sufficiently well defined and documented to proceed with further requirements analysis, decomposition, allocation, and elaboration and with further design and are ready to be baselined d. The ground segment architecture is able to meet all ground segment requirements, including functional and non-functional requirements, as demonstrated by engineering analyses, models, simulations, and prototypes e. The ground segment architecture and operational concept is consistent with the user's concept of operations f. The ground segment architecture is feasible given available technologies that are sufficiently mature, or will be sufficiently mature by PDR g. The ground segment requirements are fully allocated to the elements/subsystems, and the element/subsystem requirements are fully allocated to hardware and software items h. The bidirectional traceability from ground segment requirements through the ground segment and element/subsystem requirements is complete and demonstrates the capability of the ground segment architecture to meet the ground segment contractual requirements i. The allocation of ground element/subsystem requirements to hardware and software items is complete and consistent with the ground segment architecture j. The ground element/subsystem test planning for verification of the ground element/subsystem requirements is sufficiently well defined to guide the later	System threat assessment report (STAR) • Risk assessment • Technology development strategy (TDS) • Critical technology elements (CTEs) • Technology readiness assessment (TRA) • Test and evaluation strategy (TES) • Test and evaluation master plan (TEMP) • Acquisition strategy plan (ASP) • Intellectual property (IP) plan (documents as part of the ASP) • Systems engineering plan (SEP) • Lifecycle sustainment plan (LCSP) • Information support plan (ISP) • Program protection plan (PPP) cybersecurity plan (documented as an appendix to the PPP) Software assurance plan (documented as an appendix to the PPP) • Programmatic environment, safety, and occupational health evaluation (PESHE) • Software acquisition management plan (SWAMP) Clinger Cohen Act compliance assessment • Cost analysis requirements document (CARD) • Requests for proposal (RFPs) • Acquisition milestone documentation or acquisition decision point documentation <u>Contractor Products:</u> a. Updated Ground Segment Requirements Specification b. Ground Element/Subsystem Requirements Specifications c. System/Segment Design Document, or other document describing the

Principal Objectives	Major Review Products
development of test procedures and provides traceability of all ground element/subsystem requirements, including functional, non-functional, and interface requirements, to well-defined tests j. Ground segment development plans are feasible and sufficiently detailed for this point in the lifecycle k. The schedule and cost for implementing the ground segment architecture and the life cycle cost are within program constraints l. The ground segment development risk is at an acceptable level to proceed with further requirements and design activities	ground segment architecture to the level of hardware and software items and their interfaces d. Ground segment operational concept document e. Initial selection of commercial off-the-shelf (COTS) and reuse hardware and software f. Bidirectional traceability between the ground segment requirements and element/subsystem requirements g. Allocation of ground element/subsystem requirements to hardware and software items h. Allocation of ground segment and ground element/subsystem requirements to ground architecture components i. Ground element/subsystem test plans j. Updated ground segment test plan k. Engineering analyses, simulations, models, prototypes, and ground segment technology readiness evaluation l. Updated ground segment development plans m. Updated ground segment cost and schedule n. Updated ground segment risk list and handling plans for risks ranked medium to high

29.5.2.2.3 Software Architecture and Requirements Review (SAR)

The Ground Segment SAR is a multidisciplinary review to ensure that the software requirements and architecture are sufficiently well defined so that the ground segment design can proceed to PDR and so that the software development can proceed to additional downstream software design and development activities. Following a successful Ground Segment SAR, the overall software architecture and the software and software interface requirements should be sufficiently well defined to provide the foundation for subsequent software development activities, independent of the software development lifecycle model to be used.

At the SAR, the software architecture should be demonstrated to satisfy the software requirements, including functional, non-functional, and interface

requirements. Software engineering analyses, models, simulations, and prototypes are presented to support the satisfaction of software requirements by the software architecture, including performance and margin requirements, and to demonstrate that the software architecture will support the user's concept of operations.

In large ground systems there may be numerous software items residing on possibly different types of computer hardware. The SAR is intended to review the collection of software items as a whole, and thus reviews the overall software architecture that includes all software items, residing on all types of hardware in the ground segment, and their interfaces. Subsequent software reviews tend to focus on a single software item and/or a single software build. The SAR provides the opportunity for a global software review before the software development lifecycle delves deeply into the development of each software item.

Table 29-5. SAR Objectives

Principal Objectives	Major Review Products
a. The software requirements, including software interface requirements, and software architecture are sufficiently well defined and documented to proceed with subsequent ground segment and software requirements and design activities b. The software requirements and architectural design will meet the ground element/subsystem requirements allocated to software, as demonstrated by software engineering analyses, models, simulations, and prototypes c. The software requirements and architecture are consistent with the user's concept of operations d. The software architecture is feasible given available computer hardware and software technologies that are sufficiently mature, or will be sufficiently mature, by PDR e. The bidirectional traceability from ground element/subsystem requirements to software requirements is complete and demonstrates the capability of the	Contractor Products: a. Software requirements specifications and interface requirements specifications for all ground segment software items and their interfaces b. Software architecture description, including all ground segment hardware and software items (See Appendix H.2 of Adams et al. [1].) c. Software engineering analyses, models, simulations, and prototypes d. Computer hardware and software technology readiness evaluation e. Updated selection of COTS and reuse software products and allocation of these products to software architecture components f. Traceability between the ground segment and element/subsystem requirements and the software and software interface requirements g. Allocation of software and software interface requirements to software architecture components h. Software test plans for all ground segment software items i. Software development plan(s), including software quality assurance,

Principal Objectives	Major Review Products
software requirements to meet their allocated ground element/subsystem requirements k. The software test planning for verification of the software and software interface requirements is sufficiently well defined to guide the later development of test procedures and provides traceability of all software requirements, including functional, non-functional, and interface requirements, to well defined tests f. Software development plans are defined and mature and will be effective in implementing the ground segment software g. The software cost and schedule for implementing the software and the software lifecycle cost are within the ground segment constraints h. The software risk is at an acceptable level to proceed with further ground segment and software requirements and design activities	software configuration management, software assurance, software reliability, software safety, and software risk management plans j. Software cost and schedule estimates k. Software risk list and handling plans, including risk mitigation plans, for software risks ranked medium to high

29.5.2.2.4 Preliminary Design Review (PDR)

The Ground Segment PDR is a multidisciplinary review to ensure that the element/subsystem and hardware and software item requirements and the ground segment preliminary design are sufficiently well defined that the ground segment can proceed to CDR and that ground segment development can proceed with subsequent hardware and software design and development activities. Following a successful PDR, the ground element/subsystem requirements and the hardware and software item requirements, including interface requirements, should be mature enough to be baselined.

At a ground segment PDR, the ground segment preliminary design is demonstrated to satisfy the element/subsystem and hardware and software item requirements and to enable the user's concept of operations to be executed. Results of engineering analyses, models, simulations and prototypes are presented to support the satisfaction of requirements, including performance, margin, and specialty engineering requirements, by the ground segment preliminary design. Results are also presented to demonstrate that the ground segment preliminary design will support the user's concept of operations. At the ground segment PDR, an initial selection of commercial off-the-shelf (COTS)

and reuse hardware and software products should be presented, together with rationale for their selection. The initial layout for equipment installation at the operational facilities is presented and shown to meet the site fielding constraints (e.g., footprint, weight, power, and heating, ventilation and air conditioning [HVAC]) for each operational facility. PDR also shows that the cost and schedule constraints are understood and that the ground segment risks judged to be medium to high have adequate handling plans (including mitigation plans).

For large systems, there will usually be a system PDR and segment level PDRs, with the segment level PDRs preceding the system PDR. For a large ground segment with multiple ground elements/subsystems, there may be PDRs for each element/subsystem, especially if the elements/subsystems are being developed by different contractors.

Table 29-6. PDR Objectives

Principal Objectives	Major Review Products
a. The ground segment hardware and software item requirements are sufficiently well elaborated and defined to proceed with subsequent design and development activities and are ready to be baselined b. The hardware item and software item preliminary designs are sufficiently well defined and documented to proceed with further design and development activities c. The hardware and software item preliminary designs define the components and their interfaces to the level of detail appropriate to this point in the life cycle and for the software development life cycle model in use d. The hardware and software item preliminary designs are consistent with the ground segment and element/subsystem architecture e. The hardware and software item requirements are completely and correctly allocated to the hardware and software preliminary design components f. The hardware and software item preliminary designs are able to meet all hardware and software item	• Initial capabilities document (ICD) • Capability development document (CDD) • Capability production document (CPD) • Concept of operations (CONOPS) • Technical requirements document (TRD) [or system requirements document (SRF)] • Analysis of alternatives (AoA) • Preferred system architecture • Government reference architecture (GRA) System threat assessment report (STAR) • Risk assessment • Technology development strategy (TDS) • Critical technology elements (CTEs) • Technology readiness assessment (TRA) • Test and evaluation strategy (TES) • Test and evaluation master plan (TEMP) • Acquisition strategy plan (ASP)

Principal Objectives	Major Review Products
requirements, including functional and non-functional requirements, as demonstrated by engineering analyses, models, simulations, and prototypes g. Interface control documents or interface design documents are complete for all ground segment external interfaces and all interfaces among the elements/subsystems h. The hardware and software item preliminary designs are consistent with the ground segment operational concept and will support the user's concept of operations i. The hardware and software item preliminary designs are feasible given available technologies that are sufficiently mature, or will be sufficiently mature by CDR j. Make/buy/reuse decisions for hardware and software items and their design components are complete, well supported with rationale, and will meet the hardware and software item requirements k. Top-level hardware drawings are complete and consistent with the element/subsystem architecture and hardware item requirements and preliminary design l. Equipment layout drawings for the operational facilities are complete and meet the operational site constraints (e.g., footprint, weight, power, and HVAC) m. The bidirectional traceability from ground segment requirements through the ground element/subsystem requirements to the hardware and software item requirements is complete and demonstrates the capability of the hardware and software item requirements to meet the ground segment contractual requirements	• Intellectual property (IP) plan (documents as part of the ASP) • Systems engineering plan (SEP) • Lifecycle sustainment plan (LCSP) • Information support plan (ISP) • Program protection plan (PPP) cybersecurity plan (documented as an appendix to the PPP) Software assurance plan (documented as an appendix to the PPP) • Programmatic environment, safety, and occupational health evaluation (PESHE) • Software acquisition management plan (SWAMP) Clinger Cohen Act compliance assessment • Cost analysis requirements document (CARD) • Requests for proposal (RFPs) • Acquisition milestone documentation or acquisition decision point documentation <u>Contractor Products:</u> a. Updated ground segment and element/subsystem requirements specifications b. Ground segment hardware and software item requirements specifications c. Updated ground segment architecture description d. Updated software architecture description e. Ground segment hardware and software preliminary design descriptions f. Interface control documents or interface design documents for all external interfaces and for element/subsystem interfaces g. Updated ground segment operational concept document h. Make/buy/reuse decisions and selection of COTS and reuse

Principal Objectives	Major Review Products
n. The hardware and software test planning for verification of the hardware and software item requirements is sufficiently well defined to guide the later development of test procedures and provides traceability of all hardware and software item requirements, including functional, non-functional, and interface requirements, to well defined tests o. Ground segment and element/subsystem test procedures are sufficiently well defined at this point in the lifecycle to show that they will be able to verify the ground segment and element/subsystem requirements p. Hardware and software item development plans are feasible and are sufficiently detailed for this point in the lifecycle and are consistent with updated ground segment development plans q. The schedule and cost for implementing the ground segment preliminary design and the lifecycle cost are within program constraints r. The hardware and software item development risk is at an acceptable level to proceed with further design and development activities	hardware and software, with rationale i. Top-level hardware drawings j. Layout drawings for equipment in operational facilities, and analyses of ability to meet operational site constraints k. Bidirectional traceability between the ground element/subsystem requirements and hardware and software item requirements l. Allocation of ground element/subsystem requirements to hardware and software items m. Allocation of ground segment and ground element/subsystem requirements to ground architecture components n. Updated ground segment and element/subsystem test plans o. Initial ground segment and element/subsystem test procedures p. Hardware and software item test plans q. Engineering analyses, simulations, models, prototypes, and hardware and software technology readiness evaluation r. Updated ground segment development plans s. Hardware and software development plans t. Updated ground segment cost and schedule and life cycle cost estimates u. Updated ground segment risk list and handling plans for risks ranked medium to high

29.5.2.2.5 Critical Design Review (CDR)

The ground segment CDR is a multidisciplinary review to ensure that the ground segment detailed design is sufficiently well defined so that the ground segment can proceed with subsequent hardware and software design and development activities. Following a successful CDR, the ground segment detailed design is sufficiently mature and the hardware items can proceed with procurement or development, testing to verify that their requirements are met, and can be

deployed into the operational environment. The maturity of the software item design will depend upon the software development lifecycle model in use (see section 29.5.3 where software build level reviews are described).

At a ground segment CDR, the ground segment detailed design is demonstrated to satisfy the element/subsystem, to meet hardware and software item requirements, and to support the user's concept of operations. Final results of engineering analyses, models, simulations and prototypes are presented to demonstrate the satisfaction of requirements including performance, margin, and specialty engineering requirements by the ground segment detailed design. At the ground segment CDR, the final selection of COTS and reuse hardware and software products should be presented, together with rationale for their selection.

Ground segment development, integration, and test plans are reviewed for completeness and effectiveness; element/subsystem test plans are reviewed for adequacy in verifying element/subsystem requirements; and ground segment test procedures are reviewed for adequacy in verifying ground segment requirements. For hardware that will be manufactured, all manufacturing processes and controls are reviewed for adequacy to proceed to manufacturing. All hardware parts lists, drawings, and schematics should be completed before exiting CDR. The final layout for equipment installation at the operational facilities is presented and shown to meet the site fielding constraints for the operational facilities (e.g., footprint, weight, power, and HVAC). Plans for transitioning the hardware and software to operations are also reviewed for completeness and consistency with operational site requirements. CDR also shows that the updated ground segment cost and schedule estimates meet the program constraints. Major ground segment risks (i.e., those rated medium to high) have been retired or have adequate handling plans (including mitigation plans) for moving forward.

For large systems, there will usually be a system CDR and segment level CDRs, with the segment level CDRs preceding the system CDR. For a large ground segment with multiple ground elements/subsystems, there may be CDRs for each element/subsystem, especially if the elements/subsystems are being developed by different contractors.

Table 29-7. CDR Objectives

Principal Objectives	Major Review Products
a. The hardware item and software item detailed designs are sufficiently well defined and documented to proceed with further design and development activities b. The hardware and software item detailed designs are at a level of detail appropriate to this point in the lifecycle and for the software development lifecycle model in use c. The hardware and software item detailed designs are consistent with the ground segment and element/subsystem architecture d. The hardware and software item requirements are completely and correctly allocated to the hardware and software detailed design components, as shown by bidirectional traceability between the hardware and software item requirements and the hardware and software detailed design components e. The hardware and software item detailed designs are able to meet all hardware and software item requirements, including functional and non-functional requirements, as demonstrated by engineering analyses, models, simulations, and prototypes f. Interface control documents or interface design documents/drawings are complete for all ground segment external interfaces, all interfaces among the elements/subsystems, all interfaces among the hardware and software items, and all interfaces among detailed design components within each hardware and software item g. The hardware and software item detailed designs are consistent with the ground segment operational concept and will support the user's concept of operations	• Initial capabilities document (ICD) • Capability development document (CDD) • Capability production document (CPD) • Concept of operations (CONOPS) • Technical requirements document (TRD) [or system requirements document (SRF)] • Analysis of alternatives (AoA) • Preferred system architecture • Government reference architecture (GRA) System threat assessment report (STAR) • Risk assessment • Technology development strategy (TDS) • Critical technology elements (CTEs) • Technology readiness assessment (TRA) • Test and evaluation strategy (TES) • Test and evaluation master plan (TEMP) • Acquisition strategy plan (ASP) • Intellectual property (IP) plan (documents as part of the ASP) • Systems engineering plan (SEP) • Lifecycle sustainment plan (LCSP) • Information support plan (ISP) • Program protection plan (PPP) cybersecurity plan (documented as an appendix to the PPP) Software assurance plan (documented as an appendix to the PPP) • Programmatic environment, safety, and occupational health evaluation (PESHE)

Principal Objectives	Major Review Products
h. The hardware and software item detailed designs are feasible given available technologies that are sufficiently mature at this point in time i. Updated make/buy/reuse decisions for hardware and software items and their design components are well supported with rationale and will meet the hardware and software item requirements j. Hardware drawings and parts lists are complete and consistent with the element/subsystem architecture and hardware item requirements and preliminary design k. Updated equipment layout drawings for the operational facilities are complete and meet the footprint, weight, and HVAC constraints of the operational site(s) l. Manufacturing processes and controls for all hardware to be developed are sufficiently well defined and validated to proceed with manufacturing, and all procurements specifications are sufficiently well defined to proceed with procurement of COTS hardware m. The hardware and software test planning for verification of the hardware and software item requirements is sufficiently well defined to guide the later development of test procedures and provides traceability of all hardware and software item requirements, including functional, non-functional, and interface requirements, to well defined tests n. Ground segment and element/subsystem test procedures are sufficiently well defined at this point in the lifecycle to show that they will be able to verify the ground segment and element/subsystem requirements	• Software acquisition management plan (SWAMP) Clinger Cohen Act compliance assessment • Cost analysis requirements document (CARD) • Requests for proposal (RFPs) • Acquisition milestone documentation or acquisition decision point documentation <u>Contractor Products:</u> a. Updated ground segment, element/subsystem, and hardware and software item requirements specifications b. Updated ground segment architecture and software architecture descriptions c. Ground segment hardware and software detailed design descriptions d. Updated interface control documents or interface design documents for all external interfaces and for element/subsystem interfaces e. Interface control documents or interface design documents/drawings for all interfaces among hardware items and software items and among detailed design components within each hardware and software item f. Bidirectional traceability between the hardware and software item requirements and the hardware and software detailed design components g. Updated ground segment operational concept document h. Updated make/buy/reuse decisions and selection of COTS and reuse hardware and software, with rationale i. Hardware drawings and parts lists j. Updated layout drawings for equipment in operational facilities, and analyses of ability to meet operational site constraints

Principal Objectives	Major Review Products
o. Hardware and software item development plans are feasible and sufficiently well defined to proceed with subsequent design and development activities and are consistent with updated ground segment development plans p. Updated transition plans are sufficiently well defined to guide deployment of ground segment hardware and software and turnover to operations q. The schedule and cost for implementing the ground segment detailed design and the life cycle cost are within program constraints r. Ground segment development risks, including all hardware and software risks, have been retired or have adequate handling plans to proceed with further design and development activities	k. Manufacturing processes and controls for all hardware items/components being developed, and procurement specifications for all COTS hardware l. Updated ground segment, element/subsystem, and hardware and software item test plans m. Updated ground segment and element/subsystem test procedures n. Initial hardware and software item test procedures o. Engineering analyses, simulations, models, prototypes, and hardware and software technology readiness evaluations p. Updated ground segment development plans q. Updated hardware and software development and integration plans r. Updated transition plans s. Updated ground segment cost and schedule and lifecycle cost estimates t. Updated ground segment risk list and handling plans for risks ranked medium to high

29.5.2.2.6 Test Readiness Review (TRR)

Ground segment TRRs are held before each requirements verification test event to ensure that the contractor is ready to proceed with the test event. Each TRR is intended to ensure that the ground segment hardware and software under test, personnel, test procedures, test data and databases, schedules, and test facility and other test resources (e.g., test hardware and software, test drivers, simulators, stimulators, testbeds) are ready for testing. While the test event will focus on the verification of requirements using the Demonstration (D) and Test (T) methods, verification of requirements using the Analysis (A) and Inspection (I) test methods is also addressed, and the TRR assesses readiness for completion of requirements verification using all test methods.

TRRs are held at each level of the ground segment specification tree prior to the test events held to verify the requirements at that level. Thus, in a large ground segment, there may hardware item verification testing for each hardware item, software item verification testing (usually called software item qualification testing) for each software item, element/subsystem verification testing for each element and/or subsystem, and ground segment verification testing. Each of

these verification test events is preceded by a TRR. If the TRR shows that the contractor is not ready to hold the test event, the testing does not proceed.

Table 29-8. TRR Objectives

Principal Objectives	Major Review Products
a. Updated requirements specification(s) include all approved requirements changes and have been baselined for the item(s) under test b. The hardware and software under test has completed development and integration testing, is under configuration management, is in a documented configuration replicating the operational environment, and is ready for testing c. The updated test plan(s) address all requirements in the updated requirements specification(s) for the item(s) under test and have been approved by the contractor and acquirer d. The verification methods in the requirements specification(s) and test plan(s) are consistent for each requirement being verified, and the described tests will result in verification of the requirements allocated to each test e. The test procedures are consistent with the test plan(s), will result in the verification of the requirements allocated to each test, identify the step(s) where each requirement is verified, and are sufficiently well defined to be repeatable f. The test environment has been validated to ensure all hardware and software in the environment functions as expected, is under configuration management, and is ready for the testing g. The test procedures, together with the test data and databases, have been dry run in the test environment, all redlines from the dry run(s) have been incorporated into the test	• Initial capabilities document (ICD) • Capability development document (CDD) • Capability production document (CPD) • Concept of operations (CONOPS) • Technical requirements document (TRD) [or system requirements document (SRF)] • Analysis of alternatives (AoA) • Preferred system architecture • Government reference architecture (GRA) System threat assessment report (STAR) • Risk assessment • Technology development strategy (TDS) • Critical technology elements (CTEs) • Technology readiness assessment (TRA) • Test and evaluation strategy (TES) • Test and evaluation master plan (TEMP) • Acquisition strategy plan (ASP) • Intellectual property (IP) plan (documents as part of the ASP) • Systems engineering plan (SEP) • Lifecycle sustainment plan (LCSP) • Information support plan (ISP) • Program protection plan (PPP) cybersecurity plan (documented as an appendix to the PPP) Software assurance plan (documented as an appendix to the PPP)

Principal Objectives	Major Review Products
procedures, and all problems and discrepancies encountered during the dry runs have been documented in the problem reporting system h. Bidirectional traceability between the requirements under test, the tests in the test plan(s), the test procedure(s) applicable to the requirements, and the steps in the test procedure(s) where the requirements are verified is complete and demonstrates that the test procedures will verify the requirements under test i. A well controlled test process is defined and documented in the test plan(s) for the item(s) under test, along with detailed test schedules, roles and responsibilities, and personnel assignments j. Quality assurance presence is assured for all testing k. The hardware and software under test is sufficiently mature to begin testing, based on the severity of the open problems or discrepancies in the hardware and software under test l. No category 1 severity problems or deficiencies (that is, catastrophic or critical problems) are open on the hardware or software under test as of the TRR m. Open problems or discrepancies that affect test procedure execution or requirements verification are well understood, and workarounds exist	• Programmatic environment, safety, and occupational health evaluation (PESHE) • Software acquisition management plan (SWAMP) Clinger Cohen Act compliance assessment • Cost analysis requirements document (CARD) • Requests for proposal (RFPs) • Acquisition milestone documentation or acquisition decision point documentation <u>Contractor Products:</u> a. Configuration managed hardware and/or software item(s) under test, in a configuration replicating the operational environment b. Updated requirements specification(s) for the item(s) under test c. Updated test plans for the item(s) under test, including detailed test schedules, roles and responsibilities, and personnel assignments d. Test procedures for the item(s) under test e. Dry run result(s) for the item(s) under test f. B-directional traceability between the requirements under test, the tests in the test plan(s), the test procedure(s) applicable to the requirements, and the steps in the test procedures where the requirements are verified g. Configuration managed test environment, including test beds, test facilities, test tools, test hardware and software, test drivers, test data, and databases h. Open problem and discrepancy reports on the hardware and software under test, with assigned severity levels and documented consequences

29.5.2.3 Major Technical Reviews Lessons Learned

The following lessons learned apply to the major technical reviews:

- A compliance standard for the major technical reviews should always be placed on the contract. This standard should contain clear entry and exit criteria for the reviews so that disagreements between the acquirer and contractor do not occur. Examples are found from the IEEE and Peresztegy [1] and [10,11].

- A major technical review should not be held if the contractor has not met the entrance criteria for the review. Holding a review for which the contractor is unprepared solely to meet a schedule is detrimental to the ground segment development because the quality of downstream technical work will be adversely affected. If doubt exists as to the satisfaction of the entrance criteria, a readiness assessment by an independent team should be performed.

- The previous, now cancelled, military standard on major technical reviews, MIL-STD-1521B, should not be used. This standard rigorously imposes a waterfall life cycle model, with each major technical review conducted only once. For most ground segments this will cause contractual difficulties when trying to fit a single set of major technical reviews into an incremental development life cycle model.

- The ground segment SRR and SFR/SDR may be held once for the entire ground segment because these are high level reviews. However, the contract should specify that a SAR, PDR, and CDR should be held for each ground segment increment. The contract should specify that a TRR should be held for each verification event at each level of the ground segment specification tree.

- The contract needs to require delivery of technical data that support the major technical reviews sufficiently in advance of the reviews that the acquirer's technical personnel can thoroughly review the material before the review is conducted.

- The technical personnel need to be in attendance at the major technical reviews. This is true for both acquirer and contractor personnel so that in-depth technical discussion can be held when necessary.

- Operations personnel (e.g., users and operators) should be included as reviewers for every major technical review so that the ground segment

is properly reviewed from the operations perspective. Participation of the operations personnel will help ensure that the ground segment will be useful and will meet the challenges that operations and users have in completing their tasks and their mission. In addition, engaging the operations personnel early in the development lifecycle enables expectations about the new ground segment to be established and potential resistance of the operations community to the new ground segment to be overcome.

- Support personnel (e.g., logistics personnel, hardware and software maintainers) should be included as reviewers for the major technical reviews to ensure the ground segment is properly reviewed from a maintenance perspective. Participation of the support personnel will help ensure that the ground segment will meet its maintainability requirements and will be able to be supported after it is operational.

29.5.3 Software-specific Joint Technical Reviews

Software-specific joint technical reviews are used to delve deeply into details about the software planning, requirements, architecture, design, implementation, and testing that cannot be addressed by the higher-level major technical reviews. The intention of software-specific reviews is to have contractor and acquirer software experts review the evolving software technical products throughout the software development life-cycle. Because most ground segment software is implemented in an iterative type of lifecycle model where the software is developed in a series of builds, the software-specific joint technical reviews occur on a build-by-build basis. If the software is developed by a "once-through" lifecycle model (i.e., Waterfall), it is treated as having a single build, and each software-specific joint technical review is held only once.

All of the software-specific technical reviews do not need to be held for each build. The details of which reviews are to be held in each build, the entry and exit criteria for each review, the specific products to be reviewed, the review process to be followed, and the mechanisms for documenting action items and identified discrepancies are to be documented in the software development plan and agreed to by both the contractor and acquirer.

The overall objectives of the software-specific joint technical reviews are as follows:

- Review evolving software products, using a pre-defined set of evaluation criteria
- Review and demonstrate proposed technical solutions
- Provide insight and obtain feedback on the technical effort

- Bring to the surface and resolve technical issues
- Identify and review the status of near- and long-term risks regarding technical, cost, and schedule issues
- Arrive at agreed-upon mitigation strategies for identified risks
- Identify risks and issues to be raised at joint management reviews
- Review status of risk mitigation plans already in place, and update if necessary
- Determine if software risk mitigation is proceeding according to the updated software risk mitigation plans
- Provide for ongoing communication between acquirer and developer technical personnel

More information about the software-specific technical reviews is found in section 5.18 and Appendix E of the *Software Development Standard for Mission Critical Systems* [1].

29.5.3.1 Execution by Acquisition Phase (Software-specific Joint Technical Reviews)

The software-specific joint technical reviews described in section 29.5.3.2 occur at the software build level as shown in Figure 29-2. The software reviews generally occur as follows:

- Software Build Planning Review (SBPR)—At the completion of planning for the build

- Software Build Requirements and Architecture Review (SBRAR)—At the completion of defining the software requirements and architecture for the build

- Software Build Design Review (SBDR)—At the completion of the detailed design for the build

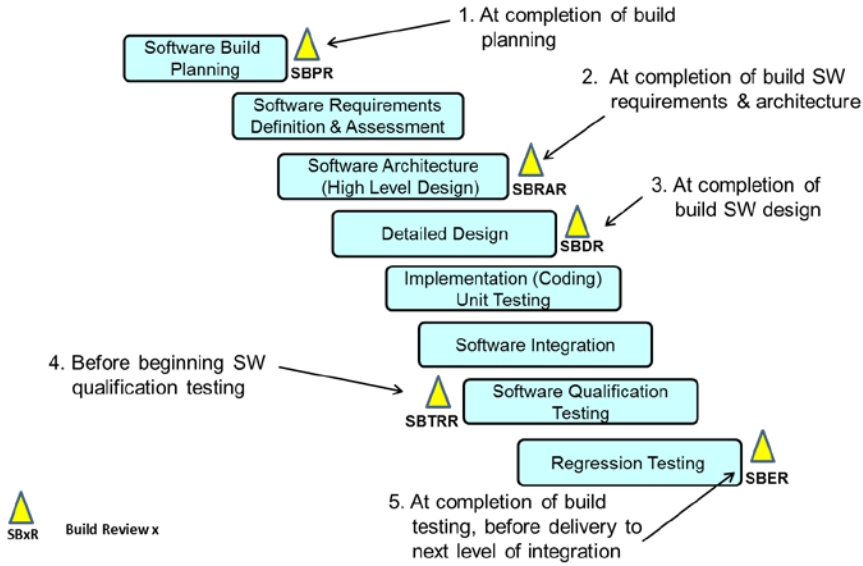
- Software Build Test Readiness Review (SBTRR)—At the start of software qualification testing for the build

- Software Build Exit Review (SBER)—At the conclusion of the build

It is not necessary for each build to have all of the software-specific joint technical reviews. The reviews that are appropriate for each build depend upon the software lifecycle model in use, the content of the build, and the future use of the build. For example, the activity of software qualification testing is frequently not held for each build, but only for those builds that will be delivered outside of the software organization (e.g., to systems engineering,

integration and test for higher-level integration). Thus, only those builds will have the SBTRR. The activity of software build planning is a major activity in the first build, and therefore should have a SBPR, and similarly for subsequent builds where there are major changes in the build planning (e.g., due to changes in requirements flowed down to software or additional deficiency/discrepancy reports assigned to the build for correction).

The software development methodology and lifecycle model in use has a large effect on the placement of the software-specific joint technical reviews. Figure 29-2 is applicable to the commonly used incremental and evolutionary software development lifecycle models, but the placement of the software reviews is significantly different for other software development lifecycle models, such as those used for agile software development and for software maintenance. The developer should address the plans for placement of the software reviews in the software development plan.



SW = software.

Figure 29-2. Software-specific joint technical reviews.

29.5.3.2 Types of Software-specific Joint Technical Reviews

29.5.3.2.1 Software Build Planning Review (SBPR)

The SBPR is held near the beginning of the build to determine whether the planning for the build is adequate and the resources needed for the build will be

available. For top level objectives of the SBPR, see Table 29-9. For detailed objectives of the SBPR, see Appendix E.3.1 of Adams [1].

Table 29-9. SBPR Objectives

Principal Objectives	Major Review Products
a. The requirements and architectural design are sufficiently mature, to plan builds b. The software processes are sufficiently defined, mature and effective for performing the build software development activities and are suitable for the program scope and complexity c. The build strategy is suitable for the program scope and complexity and for meeting the needs and timing of the higher integration levels (e.g., hardware integration, element and subsystem integration, segment integration, system integration) d. The resources for the builds are allocated and sufficient for each build and for supporting simultaneous build development and operations support e. All software risks ranked medium to high are properly identified with adequate risk handling plans, and all risk mitigation is proceeding as planned	Contractor Products: a. Software development plan (SDP), software standards and procedures (work instructions) b. Software master build plan (SMBP), including allocation of requirements, functionality, and architectural components to builds and changes to the SMBP since its last update c. Build schedules and staffing plan d. Estimated number of source lines of code (SLOC) per build and other metrics e. Discrepancy reports (DRs) and the allocation of the DRs to the builds for fixing f. Integrated master plan (IMP) and integrated master schedule (IMS) g. Software and higher level requirements, software and external interface requirements, software architecture, software and higher level test plans, to the extent these are completed at the time of the SBPR h. Updated software risks and status of risk handling efforts for software risks ranked medium to high

29.5.3.2.2 Software Build Requirements and Architecture Review (SBRAR)

The SBRAR is held at the completion of defining the software requirements (including software interface requirements), software architecture, and software test plans for the software in the build. For top level objectives of the SBRAR, see Table 29-10. For detailed objectives of the SBRAR, see Appendix E.3.2 of Adams [1].

Table 29-10. SBRAR Objectives

Principal Objectives	Major Review Products
a. The software requirements and architectural design are adequate for meeting the higher- level requirements allocated to software b. The software requirements and architectural design are sufficiently mature to proceed with dependent software and system development activities c. The software processes are sufficiently defined, mature, and effective for developing the software needed to meet system requirements and operational needs, and are suitable for the program scope and complexity d. The software test plans are sufficiently robust to ensure thorough nominal and off-nominal testing of the software products to demonstrate that the software requirements are verified in the target environment e. The software development and test environments are established and have adequate capability and capacity to meet the software development and verification requirements and schedules f. The software requirements, architectures, qualification test plans, and the master software build plan are correct, consistent, complete, and traceable, and are supported by engineering analyses g. All software risks ranked medium to high are properly identified with adequate risk handling plans, and all risk mitigation is proceeding as planned	<u>Contractor Products:</u> a. Software requirements specifications (SRSs) and interface requirements specifications (IRSs) for the software in the build b. Higher level requirements allocated to software, including all changes c. Interface control documents (ICDs) or interface requirements specifications (IRSs) for interfaces external to the build d. Software architecture description for the software e. Top-level computer system hardware and software architecture from the system/segment architectural design f. Ground segment operational concept document g. Make/buy/reuse decisions and selection of COTS and reuse software, with rationale and software technology readiness evaluation for the build h. Software test plans (STPs) for verifying the software requirements and software interface requirements being satisfied by the build i. Bidirectional traceability: • Between higher-level requirements allocated to software and software requirements and software interface requirements allocated to the build • Between software and software interface requirements allocated to the build and software architecture components for the build • Between software and software interface requirements allocated to the build and tests in the software test plans for verifying those requirements

Principal Objectives	Major Review Products
	j. Software engineering analyses, models, trade studies, simulations, or prototypes k. Updated software development plan l. Updated software master build plan m. Updated software cost and schedule and lifecycle cost estimates n. Latest software metrics reports o. Updated software risks and status of risk handling efforts for software risks ranked medium to high

29.5.3.2.3 Software Build Design Review (SBDR)

The SBDR is held at the completion of the software detailed design for the build. For top-level objectives of the SBDR, see Table 29-11. For detailed objectives of the SBDR, see Appendix E.3.3 of Adams [1].

Table 29-11. SBDR Objectives

Principal Objectives	Major Review Products
a. The software design is adequate for meeting the software requirements (including interface requirements) allocated to this build and is consistent with the system and software architectures b. The software design is sufficiently mature to proceed with the build's dependent software implementation, integration, and test activities c. The software processes (including coding and testing standards) are sufficiently defined, mature and effective for developing the software needed to meet the requirements, and are suitable for the program scope and complexity d. The software qualification test plans and cases are sufficiently robust to ensure thorough nominal and off-nominal testing of the software product to demonstrate that the software requirements are verified in the target environment	Contractor Products: a. Software detailed design, database design, and interface design descriptions (software design document [SDD], database design document [DBDD], interface design document [IDD]) b. Software integration test plans and descriptions c. Software test cases in the software test description (STD) d. Other software-related information, including any updates since SBPR and SBRAR: • SRSs, IRSs, ICDs • Software architecture description • STPs • SMBP • Bi-directional traceability among software requirements, architecture, and tests • Higher level requirements allocated to software

Principal Objectives	Major Review Products
e. The software development and test environments are established and have adequate capability and capacity to meet the software development and verification requirements and schedules, including multiple concurrent builds f. The software requirements, architecture, design, qualification test plans and cases, and the master software build plan are correct, consistent, complete, and traceable, and are supported by engineering analyses g. All software risks ranked medium to high are properly identified with adequate risk handling plans, and all risk mitigation is proceeding as planned	• Ground segment operational concept document (OCD) • IMP, IMS • Discrepancy reports (DRs) allocated to the build for fixing • SDP • Standards and procedures (work instructions) especially coding standards e. Bi-directional traceability between: • Software and software interface requirements and the software design components to which they are allocated • Software and software interface requirements and the test cases in the software test description • Software architecture components and software design components f. Software engineering analyses, models, simulations, or prototypes g. Updated make/buy/reuse decisions and selection of COTS and reuse software, with rationale, and software technology readiness evaluation for the build h. Updated software size, effort, cost, schedule, and staffing estimates, and lifecycle cost estimates i. Latest software metrics reports j. Updated software risks and status of risk handling efforts for software risks ranked medium to high

29.5.3.2.4 Software Build Test Readiness Review (SBTRR)

The SBTRR is held before software qualification testing begins for those builds in which requirements verification takes place. For top-level objectives of the SBTRR, see Table 29-12. For detailed objectives of the SBTRR, see Appendix E.3.4 of Adams [1].

Table 29-12. SBTRR Objectives

Principal Objectives	Major Review Products
a. The software under test is sufficiently mature to begin the formal qualification test event b. The software qualification test plans, cases, and procedures are correct, consistent, complete, and traceable to the software and interface requirements c. The software qualification test environment is ready for the formal qualification test event to begin d. The software qualification test procedures have been successfully dry run in the software qualification test environment using established test data and were able to achieve their expected results and verify their allocated software and interface requirements e. The software qualification test procedures include both nominal and off-nominal conditions, and both nominal and off-nominal conditions have been successfully dry run with the expected results f. Disciplined test processes are in place, and all necessary test resources (including test personnel and the qualification test environment) are available g. All software risks ranked medium to high are properly identified with adequate risk handling plans, and all risk mitigation is proceeding as planned	Contractor Products: a. Software under test b. Software test environment c. Updated SRSs, IRSs, ICDs d. Updated STPs e. STDs with updated test cases and completed test procedures f. Test data and databases g. Dry run results h. Open software discrepancy reports i. Software version description (SVD) j. Software user's manual (SUM) or other documentation containing user/operator instructions k. Updated software risks and status of risk handling efforts for software risks ranked medium to high

29.5.3.2.5 Software Build Exit Review (SBER)

The SBER is held before the build is declared complete. For top level objectives of the SBER, see Table 29-13. For detailed objectives of the SBER, see Appendix E.3.5 of Adams [1].

Table 29-13. SBER Objectives

Principal Objectives	Major Review Products
a. The updated software requirements, architecture, and design remain valid, complete, consistent, stable, and traceable b. The software qualification test plan, cases, and procedures are robust, valid, complete, consistent, stable, traceable, and executable c. The software build integration and build regression test plans and procedures are correct, consistent, complete, traceable, and have been successfully executed d. All software build integration and build regression test results are captured in test report documentation e. The software qualification test procedures were executed successfully (if applicable to the build under review) f. All software qualification test results are captured in the STRs g. Software requirements verification status is maintained h. All SRS and IRS requirements that were scheduled for verification in this build, but were not verified, have been assigned to a future build for completion i. All problems have been documented as DRs and categorized for severity j. All severity 1 and 2 problems have been resolved, retests have been successfully executed, and the results have been documented k. All open severity 3 problems have been dispositioned and assigned to future builds for resolution l. The software is sufficiently mature for the build to be considered completed m. Updated software size, effort, cost, schedule, and staffing estimates remain consistent with the build	Contractor Products: a. Updated software products: • SRSs, IRSs • Software architecture and detailed design documents • STPs, STDs • SVD • SUM b. Software test reports (STRs) containing results of qualification testing for the build, if applicable c. Software build integration and regression test plans and procedures, and execution results d. Software requirements verification status e. Open software discrepancy reports, with severity categorization, disposition, and assignment to future build for resolution f. Software product specification (SPS) and other software maintenance documentation, if applicable g. Updated SMBP h. Updated software size, effort, cost, schedule, and staffing estimates, and lifecycle cost estimates i. Latest software metrics reports j. Updated software risks and status of risk handling efforts for software risks ranked medium to high

Principal Objectives	Major Review Products
results, and the updated SMBP remains executable for future builds n. All software risks ranked medium to high are properly identified with adequate risk handling plans, and all risk mitigation is proceeding as planned	

29.5.3.3 Lessons Learned (Software-specific Joint Technical Reviews)

The following lessons learned apply to the software-specific joint technical reviews:

- The software-specific joint technical reviews should be shoulder-to-shoulder reviews with the acquirer and contractor software technical personnel. Actual software artifacts should be reviewed rather than having the contractor prepare briefing charts. Participants at the reviews should have enough software expertise to review the software artifacts effectively.

- *The Software Development Standard for Mission Critical Systems* [1] should be specified on the contract as a compliance document. This standard contains the objectives of the software-specific joint technical reviews.

- The contract should specify that a Software Development Plan (SDP) be delivered, with acquirer approval required. The SDP should be required to document the placement of the software-specific joint technical reviews in the contractor's software build structure. The contents of the SDP are specified in Appendix H.1 of [1].

- The acquirer and contractor should agree to detailed entrance and exit criteria for each software-specific joint technical review far enough in advance of the review that supporting materials can be prepared by the contractor and thoroughly reviewed by the acquirer's software technical experts before the review.

29.5.4 Deployment and Operations Reviews

Deployment and Operations reviews are held to verify that the ground segment is available and ready for shipment, installation, verification, validation, and transition to operations. The key ground segment deployment and operations

reviews include: Pre-Ship Review (PSR), Segment Completion Review (SCR), Operational Readiness Review (ORR) and Operational Acceptance Review (OAR). Pre-Ship Audit (PSA) and Consent to Ship (CTS) review will be discussed as part of the PSR process. The following three enterprise level reviews will be briefly discussed, but only from the ground segment perspective: Enterprise Readiness Review (ERR), Flight Readiness Review (FRR) and Launch Readiness Review (LRR).

29.5.4.1 Execution by Acquisition Phase (Deployment and Operations Reviews)

The key deployment and operations reviews described in section 36.5.4 occur at the ground segment and system levels. These reviews generally occur as follows:

- Pre-Ship Review (PSR) – at the completion of factory verification activities, when the ground segment component(s) are ready to be shipped to the operational site
    - Pre-Ship Audit (PSA) – typically within the month prior to PSR
    - Consent to Ship (CTS) – typically at the completion of a successful PSR; may occur concurrently with PSR
- Segment Completion Review (SCR) – at the completion of site installation, integration, verification, and validation activities
- Operational Readiness Review (ORR) – at the completion of system level verification and validation activities, in preparation for transition
- Operational Acceptance Review (OAR) – prior to Initial Operational Capability (IOC) and Final Operational Capability (FOC)

The following three reviews are held to assess the readiness of all of the segments of the enterprise (i.e., spacecraft, ground segment, and launch vehicle) to support a launch campaign. This chapter will focus on the ground segment activities required to support these readiness reviews.

- Enterprise Readiness Review (ERR) – at the completion of a successful space segment PSR, in preparation for the spacecraft to ship to the launch site

- Flight Readiness Review (FRR) – typically occurs just days prior to launch to ensure that the space, ground, and user segments are ready to support the mission

- Launch Readiness Review (LRR) – typically occurs just days prior to launch to ensure that the space and launch vehicles are ready to support the launch and mission; the ground segment is polled to assure that it is also ready to support the mission after separation from the launch vehicle.

29.5.4.2 Types of Deployment and Operations Reviews

29.5.4.2.1 Pre-Ship Review (PSR)

Pre-Ship Review (PSR) is conducted to ensure the readiness of ground hardware and software to be shipped from the factory and installed at the operational site. PSR evaluates whether all formal testing has been completed, mitigation plans are in place for open items, installation and checkout procedures and training/operational documentation have been developed and approved, and resource planning has been completed and approved. The objectives of the PSR are shown in Table 29-14.

Table 29-14. PSR Objectives

Principal Objectives	Major Review Products
a. Ground segment product list, configuration items and other shipment documentation is complete, baselined, and approved. b. All components for shipment have been identified and are ready for shipment. c. All ground segment baseline changes approved since CDR are documented. d. Formal factory testing has been successfully completed. e. PSA has been successfully completed; checklist and findings have been documented. f. All DRs have been documented and categorized for severity. g. Mitigation plans or workarounds for open DRs have been documented and approved.	a. Site Installation request for change (RFC) b. Ground Segment Requirements Verification Reports c. DRs with documented and approved mitigation or workaround plans d. Shipping Plan and Procedures e. Installation Plan and Procedures f. Site Integration and Verification Plan g. PSA Report or results

Principal Objectives	Major Review Products
h. Shipping plan and procedures, including mitigation plans for shipping and handling issues, is approved and in place. i. Installation plan and procedures, including support for installation issues, is approved and in place. j. Security and Information Assurance certifications have been approved and received. k. The shipping and receiving sites are ready to support the shipment; appropriate resources at both shipping and receiving sites have been coordinated and shipment issues mitigated.	

A Pre-Ship Audit (PSA) is usually conducted to verify that all configuration and shipment-related items have been addressed prior to shipment. The PSA checklist confirms that the contents and configuration of the product(s) to be shipped are complete, baselined, and documented. PSA deliverables include the Product Ship List and the Functional and Physical Configuration Audit (FCA/PCA) Reports. PSA findings are typically reported at the PSR. If PSA findings uncover critical issues that need to be addressed, PSR may be postponed until the issues are worked off.

Consent to Ship (CTS) review is conducted to ensure that the operational sites are ready to receive the ground deliverables in preparation for system and/or enterprise integration and test. All impacted segments and external systems are polled for readiness to support system and/or enterprise level testing after the ground deliverables are installed and verified at site(s). This review confirms that all critical actions from PSR have been mitigated or have approved mitigation plans in place.

29.5.4.2.2 Ground Segment Completion Review (SCR)

Ground SCR is conducted to ensure that the ground segment components have been successfully delivered, installed, and verified at the site(s) and that the ground segment is ready to support system-level verification and validation activities. It also verifies that the ground segment requirements have been fully satisfied, and, if not, that the risk has been successfully mitigated in preparation for system verification and validation. If system-level verification/validation is not required for the installed ground segment component(s), the completion of this review will initiate transition to operations activities.

Table 29-15. Ground SCR Objectives

Principal Objectives	Major Review Products
a. All deliverables (as documented in the product list), configuration items, and shipping documentation have been received at site b. Site installation and integration activities have been successfully completed c. Site verification and validation activities have been successfully completed, including external interfaces d. Segment and site requirements have been fully satisfied/verified e. Training has been documented and approved f. Maintenance plans and procedures and appropriate resources have been approved and are in place g. Mitigation plans and/or workarounds for open DRs are documented, accepted and approved h. Security and Information Assurance documentation and certifications are approved, current and in place i. New/updated functionality is available and the ground segment is stable and ready to support system verification/validation or transition activities	a. Ground Segment Requirements Verification Reports b. DRs with documented and approved mitigation or workaround plans c. Site Integration and Verification Results

29.5.4.2.3 Operational Readiness Review (ORR)

ORR is conducted to ensure that the ground deliverables have been successfully installed and verified, and that the system is stable and ready to successfully transition to operations. All sites, staff, equipment, and operational plans and procedures are ready and available to support the transition. Successful completion of the ORR initiates transition activities.

Table 29-16. ORR Objectives

Principal Objectives	Major Review Products
a. Transition documentation is complete, baselined, and approved b. Necessary system and external resources (e.g., staffing, systems, networks, etc.) are available and ready to support transition and operations c. Configurations are documented, baselined, and approved d. Transition and maintenance plans are documented, baselined, and approved e. Training documentation has been documented, baselined, and approved, and training resources are ready to support transition f. Maintenance plans and procedures and appropriate resources have been approved and are in place g. Mitigation plans and/or workarounds for transition-related open DRs are documented, accepted, and approved h. Security and Information Assurance documentation and certifications are approved, current, and in place i. Transition risks have been identified, documented, and accepted by all stakeholders	a. Transition RFC b. Operational Readiness Plan c. Network Interface Plan d. Installation Plan and Procedures e. System Transition Plans and Procedures f. Segment Transition Plans and Procedures g. Site Transition Plan and Procedures h. Ground SCR report or Installation Completion Report i. DRs with documented and approved mitigation or workaround plans j. Security Accreditation Package, including but not limited to System Security Plan, Security Assessment Report, Authorization to Operate k. Training Documentation and appropriate Certifications l. Operational Tech Orders and Workarounds m. User Guides n. Maintenance Manuals (software, hardware, firmware, COTS) Functional and Physical Configuration Audits (checklist or report)

29.5.4.2.4 Operational Acceptance Review (OAR)

OAR is conducted to ensure that the ground segment and/or system is mature enough to be approved as the new baseline. Operational and end users evaluate the new/updated functionality and its impact on the performance of the system, enterprise, and/or associated external systems over a predetermined time period to determine whether it meets user needs and/or expectations. As noted in section 29.5.4.1, OAR is typically conducted at IOC and FOC to ensure that the new ground segment meets requirements and expected performance before being designated as the new baseline.

Table 29-17. OAR Objectives

Principal Objectives	Major Review Products
a. To ensure that the following are in place to support an IOC or FOC declaration: • Baselined and approved product list • Baselined and approved configuration, including system, facility, and communications components • Roles and responsibilities of operational and external stakeholders and end users • Adequate staffing for operations and maintenance • Approved security and information assurance (IA) requirements and certifications/accreditations b. Mitigation plans and/or workarounds for transition-related open DRs are documented, accepted and approved c. Transition risks have been identified, documented, and accepted by all stake- holders (operational, external, end user) d. Operational, external and end user organizations have accepted and are ready to transition to the initial or full segment/system capabilities	a. Transition RFC b. Ground SCR report or Installation Completion Report c. Transition Activity Reports d. Functional and Physical Configuration Audits (checklist or reports) e. DRs with documented and approved mitigation or workaround plans

29.5.4.2.5 Enterprise Readiness Review (ERR)

ERR is conducted to verify that all enterprise components, including all ground segments, are ready to support the spacecraft's shipment to the launch site, integration and verification activities in preparation for launch, and activities in support of launch, initialization and transition to operations. All enterprise stakeholders (e.g., launch, ground, communications, operations, external systems/interfaces) confirm their readiness to support launch-related activities and enterprise level testing.

The objectives of the ERR relative to the ground segment are shown in Table 29-18.

Table 29-18. ERR—Ground-Specific Objectives and Products

Principal Objectives	Major Review Products
a. Ground segment is available, and all necessary resources are ready to support spacecraft shipment to launch site, launch-related activities, initialization, and transition to operations b. Mitigation plans and/or workarounds for open ground segment DRs that impact launch, initialization, or operational transition are documented and approved	a. Ground-related DRs (impacting launch, initialization, and transition activities) with documented and approved mitigation or workaround plans b. Ground Segment Resource Plan c. Ground Segment/Site Verification Plan and Results

29.5.4.2.6 Flight Readiness Review (FRR) and Launch Readiness Review (LRR)

FRR and LRR are conducted to ensure that all ground software, hardware, procedures, and personnel are ready to support launch, transition, and on-orbit activities (testing, deployment, handover). FRR/LRR assess the thoroughness of ground systems test and analysis, personnel training, operational demonstration, and hardware certification. FRR/LRR certifies that all segments of the system are properly prepared to support the program into an operational mode.

Table 29-19. FRR/LRR—Ground-Specific Objectives and Products

Principal Objectives	Major Review Products
FRR a. Present evidence to government launch authority and program office (PO) of the space-flight worthiness and readiness of all segments to support launch LRR a. Present evidence to government launch authority, government user organization, and PO of readiness to launch	FRR a. Briefing from space and ground segments and launch site b. Launch site detachment flight certification c. Space-flight worthiness certification LRR a. Launch go/no-go letter

29.5.4.3 Lessons Learned (Deployment and Operations Reviews)

The following lessons learned apply to the key deployment and operations reviews:

- Entry and exit criteria, in addition to roles and responsibilities, should be documented and understood by all stakeholders.

- Deployment and operations reviews should not be conducted if entry criteria have not been met. Holding a review before all resources/inputs are ready and available results in milestone review postponements or delta reviews, causing delays in the deployment of the ground capabilities.

- Resource planning for ground support and maintenance should be documented and approved as early as possible.

- Securing the necessary approvals for security certifications/accreditations should be initiated prior to the review to ensure receipt or closure plan is in place by the review.

- All stakeholders (operations, external, and end users) should be included in deployment and operations reviews as early as possible to ensure that their needs are appropriately addressed, limitations are considered, and unreasonable requirements are not placed on the users due to limitations and/or discrepancies. Creating undue burden on the users may result in non-acceptance of the new baseline.

- When presenting at enterprise level reviews, presenters should understand the expectations of the government leadership for the briefing. Additional information, beyond what is briefed in a normal readiness review is expected. The specific level and types of information may vary from different government authorities.

29.5.5 Independent Reviews

Independent reviews are initiated by contractor or acquirer management when an objective assessment of specific program issues is needed for decision-making. Using a team of qualified technical people having no association with the program under review is a common technique for ensuring the results of the review will be objective and not biased by the official position of the program's management or technical personnel. In addition, a team of people who are independent of the program under review will ensure the team will not be constrained by fear of reprisal by their management.

There are a number of types of widely used independent reviews. This chapter addresses the Independent Program Assessment (IPA), Independent Readiness Review Team (IRRT), and Independent Review Team (IRT). In addition, three

types of software independent assessments are discussed: software architecture evaluations, software process appraisals, and software readiness assessments.

29.5.5.1 Execution by Acquisition Phase (Independent Reviews)

Independent assessments/reviews occur throughout the acquisition lifecycle.

IPAs are the assessment of the government program office's readiness to proceed into the next acquisition phase. IPAs may occur before any major milestone and are intended to prevent hidden cost overruns, schedule slips, or performance reductions.

IRRT reviews customarily start during the integration and test (I&T) phase of the program. The purpose is to assure that any problems that occur during I&T are understood and appropriately resolved. The IRRT main function is to assess the problem resolutions and determine if there is any residual risk to the overall program mission. The IRRT risk assessment does not take in consideration any cost or schedule impact.

IRTs can be performed at any program phase, including design reviews, integration, and test prior to beginning operations or when the program encounters technical, schedule, or cost difficulties. An IRT review may be requested by the government program office, contractor, The Aerospace Corporation, or a responsible independent government agency. The IRT is made up of members whose skill set is determined by the nature of the review. The IRT members will determine the preferred method to accomplish the review (interviews, document reviews, software analysis). The IRT normally provides interim progress reports in addition to the final report.

Software independent assessments may occur throughout the software lifecycle as described below. Some may be initiated prior to a major technical review (such as SDR/SFR, SAR, or PDR) or major event (such as turnover of software to systems integration). Others occur as contractually required or as specific program circumstances dictate.

The first software architecture evaluation is held for the SDR/SFR. The system architecture is defined at that point in time and contains the software architecture to the level of the major software components (e.g., software items). The next software architecture evaluations are held for the SAR and PDR, where the complete software architecture across and within each software item should be defined. Software architecture evaluations may be held thereafter if significant changes in the software architecture occur (e.g., for changes in the ground segment requirements allocated to software).

The initial software process appraisal is usually held at the beginning of the ground segment contract following the contractor's completion of the software development plan and tailoring of the corporate software processes for the program. Software process appraisals are recommended to be held at regular intervals throughout the software development lifecycle (e.g., every one or two years). A software process appraisal does not need to cover all process areas but may focus on those process areas most in use as time progresses through the life cycle.

Software readiness assessments are held to determine the readiness of the software for the next phase of development or for an upcoming designated event, including the associated risk of proceeding. Software readiness assessments may thus, for example, be held before each major technical review or before turnover of the software to systems engineering for integration at a higher level. The number and placement of software readiness assessments in the software development lifecycle should be determined based on those places judged to be of the highest risk for proceeding.

29.5.5.2 Types of Independent Reviews

29.5.5.2.1 Independent Program Assessment (IPA)

IPAs are conducted before each milestone and assess the readiness of the Program Office to proceed to the next acquisition phase. IPAs are systematic reviews of major functions and duties that POs are expected to complete to accomplish their acquisition strategies. IPAs provide recommendations to the POs to prevent hidden cost overruns and schedule slips, and resolve deficiencies to facilitate successful milestone review boards.

Table 29-20. IPA Objectives

Principal Objectives	Major Review Products
Ensure the adequacy of the ground segment in the following areas: a. Technology Readiness Assessment • Systematic, metrics-based process and accompanying report that assesses the maturity of critical hardware and software technologies to be used in systems b. Acquisition Strategy and Approach c. Source Documents (past acquisition decision memorandums [ADMs], Congressional Input)	a. IPA Lessons Learned b. Summary of IPA Assessments c. IPA Final Briefing

Principal Objectives	Major Review Products
d. Capability Need (analysis of alternatives [AoA], preferred alternative, concepts of operation [CONOPS], technical requirements document [TRD]) e. Top Level Schedule f. Interdependency and Interoperability (Architectures) g. Risk and Risk Management h. Technology Maturation i. Systems Engineering Approach (software engineering plan [SEP], programmatic environmental safety and health evaluation [PESHE]) j. Software Approach k. Industrial Capability and Manufacturing Readiness l. Business Strategy/Data Management Strategy m. Resource Management (Staffing, Cost Drivers, program office estimate [POE]/spacecraft control processor [SCP], Funding Profile, earned value management [EVM]) n. Program Protection Planning (cost performance index [CPI], program protection plan [PPP], information assurance standards [IAS]) o. Test and Evaluation (Developmental and Operational Testing) p. Life Cycle Sustainment Plan q. Clinger-Cohen Act r. Execution Status and Other s. Acquisition decision memorandum[ADM] Recommendations	

29.5.5.2.2 Independent Readiness Review Team (IRRT)

The IRRT provides government leadership with a residual technical risk assessment prior to a major milestone (mission readiness review mission readiness review [MRR], FRR/LRR). The IRRT uses a set of risk criteria (probability and consequence) to evaluate the mission assurance risk of major issues or problems. The risk assessment is independent from that of the government program office and contractor. The team is made up of government, The Aerospace Corporation, system engineering and technical assistance (SETA), and systems engineering and integration (SE&I) personnel. The IRRT

reviews only the mission assurance risks of the program; cost and schedule are not considered.

Table 29-21. IRRT Objectives

Principal Objectives	Major Review Products
a. Determine scope, objectives and schedule for IRRT review b. Determination of selected critical areas to review c. Hardcopy/softcopy data collection • Obtain relevant documents, including repositories • Review background documentation and information • Determine and request appropriate contract data requirements list (CDRL) items d. Initial "interview" data collection • Determine and speak with selected key stakeholders (contractors, user organization personnel, operators, and program office personnel) e. For critical areas, perform review and analysis activities which may include, but are not limited to, the following: • Review and analysis of selected past testing (e.g., formal qualifications test [FQT], system testing, integration testing) and current test plans including proposed operational testing, and review of selected test artifacts (e.g., test plans, test procedures, test reports) • Review of code stability data (e.g., rate of DR generation) for selected areas • Review of selected test support software and hardware (e.g., simulators, testbeds) • Possible "non-obtrusive" test witnessing • Review and analysis of selected CRDL items	a. Generate findings of each review item and determine risk to mission (probability and consequence) of each risk. b. Based on the severity of the risks, provided risk reduction recommendations c. IRRT out brief to management **IRRT Documents to Review (Examples)** a. Operational testing plan b. Operational test team certification requirements for operational readiness c. System PDR/CDR slides d. Definition of Critical System and Software Requirements e. Latest versions of specifications f. Integration test plan g. Systems requirements test (SRT) and system test plans h. SRT Quick Look Report i. Software Requirement Specifications used for software item qualification testing (SIQT) j. SIQT Test Plans/Procedures k. Latest System Specification l. Latest Software Test Reports m. (Contractor) System Level Test Report n. Operational Plans o. Risk Management Plan p. Training Documents q. Definition of New, Modified or Reused Software items r. Enhanced phase operations transition (EPOT) Test Plan/Procedures s. Contingency Test Documentation t. Operations Checklist/Procedures u. Data Base Verification Procedure

Principal Objectives	Major Review Products
• Review scenarios to be run during operational utility evaluation (OUE) • Review and analysis of operator and transition team readiness • Review and analysis of transition and turnover plans • Review of maintenance and logistic readiness • Other review activities (e.g., selected review of software quality, traceability, configuration management) f. Status of security accreditation • Assess status of satisfaction of security requirements g. Status of COTS, tools integration and telemetry, tracking, and commanding (TT&C) system • Review performance of integrated system • Review plans for COTS upgrades and maintenance • Adequacy of documentation h. Review history (trends) of DRs for TT&C	v. List and a copy of all open Category 1 & 2 PR's from SIQT and System Test w. List and a copy of all open Category 1 & 2 PRs from SRT testing x. System Security Accreditation Agreement (SSAA) y. User requirements document (CDD, or equivalent) z. User CONOPS aa. Test and Evaluation Master Plan (TEMP) bb. Maintenance check list/procedures cc. Simulator Validation Plan dd. Scenarios (system, end-to-end, day-in-the-life, stress/load test)

29.5.5.2.3 Independent Review Team (IRT)

The IRT can be used at any program phase (i.e. design reviews, integration and test, program cost and schedule problems). The particular charter can be written by program management or any stakeholder authorizing an IRT. The team members are selected based on their areas of expertise that match the requirements of the charter who do not have a day-to-day involvement in the program execution. The team will determine the review method (analysis, interviews, document reviews, etc.) and the type of final product (report or presentation).

Table 29-22. IRT Objectives

Principal Objectives	Major Review Products
Ensure the adequacy of the ground segment in the following areas: a. Systems • Systems Engineering and Program Processes • Risk Management • Issues Management • Configuration Management • Schedule • Earned Value Management • Threat Preparation (back-up systems) b. Hardware/Software • Maintenance Management • Requirements Management o Traceability o Testability o Verification methods • Interface • Configuration Management • COTS Managements • Mission Data Processing and Dissemination • Operations Management	IRT Products a. IRT Findings b. IRT Observations c. IRT Recommendations d. IRT Notes and Lessons Learned e. IRT Final Briefing

29.5.5.2.4 Software-specific Independent Assessments

There are three types of software-specific independent assessments that are considered best practices for ground segment software acquisition: software architecture evaluations, software process appraisals, and software readiness assessments. The proper application of all of these assessments requires that they be written into the ground segment development contract. These independent assessments may be carried out by a government team, a contractor team, or a joint team.

29.5.5.2.4.1 Software Architecture Evaluation

A software architecture evaluation is an evaluation of the contractor's software architecture at a suitable point in the software lifecycle. The purpose of a software architecture evaluation is to determine whether the software architecture will satisfy the ground segment requirements allocated to software and whether it will enable the user's concept of operations to be executed. Software architecture evaluations can be beneficial because they often result in

clarified non-functional requirements, improved architecture documentation, early identification of risks, early evaluation of alternative candidate architectures, and productive communication among the various stakeholders. Candidate evaluation points are the SDR/SFR and SAR for the overall software architecture; SAR, PDR, and SBRAR for the software architecture of each software item; and CDR and SBDR for any software architectural updates. An architecture evaluation can focus on specific topics of importance to the program, such as software security or software supportability. The Aerospace Corporation has defined a method for evaluating software architectures that is based on a framework containing a large number of questions about the software architecture that are tailored for the specific evaluation [13]. The Software Engineering Institute (SEI) has also defined a software architecture evaluation method, the Architecture Tradeoff Analysis Method[SM] [14].

Another method of software architecture evaluation uses modeling. One example is dynamic modeling using discrete event simulation or another dynamic technique to be able to understand the execution behavior of the software architecture and to determine whether the performance requirements allocated to software will be met. Another example is to model the software reliability and availability based on the software architecture to ensure the reliability and availability requirements allocated to software will be met.

Table 29-23. Software Architecture Evaluation Objectives

Principal Objectives	Major Review Products
a. Determine whether the software architecture will satisfy the ground segment requirements allocated to software b. Determine whether the software architecture will enable the user's concept of operations to be executed c. Clarify non-functional requirements d. Improve architecture documentation e. Identify risks early in the life cycle f. Evaluate alternative candidate architectures early in the lifecycle g. Enable productive communication among the various stakeholders	Contractor Products: The products depend upon where in the lifecycle the evaluation is performed. See software architecture-related products under SFR/SDR (Table 29-4), SAR (Table 29-5), PDR (Table 29-6), CDR (Table 29-7), SBRAR (Table 29-10), and SBDR (Table 29-11).

29.5.5.2.4.2 Software Process Appraisal

A software process appraisal is an evaluation of the contractor's documented software processes and of the evidence of their use on the current ground segment development effort and on other similar programs. A software process appraisal is usually performed at the beginning of a development contract and at

regular intervals thereafter (e.g., every one to two years). It may also be performed as part of source selection. The purpose of performing a software process appraisal is to determine whether the contractor has defined documented disciplined software development processes that produce repeatable results, are consistent with the SEI's Capability Maturity Model® Integration℠ for Development (CMMI®-DEV), and are used by the entire team throughout the period of performance [15]. A software process appraisal can identify and mitigate software risks that may affect software cost, schedule, and product quality. The SEI has also defined a standard method for performing a process appraisal, the Standard CMMI®-based Appraisal Method for Process Improvement (SCAMPI℠) [16].

Table 29-24. Software Process Appraisal Objectives

Principal Objectives	Major Review Products
a. Determine whether the contractor has defined and documented disciplined software development processes that: • Produce repeatable results • Are consistent with the SEI's CMMI®-DEV • Are used by the entire team throughout the period of performance b. Identify and mitigate software risks that may affect software cost, schedule, and product quality.	Contractor Products: a. Organizational process definitions b. Organizational process definitions tailored for the program c. Software development plan d. Software standards and procedures (work instructions) e. Evidence of use of all processes on current program and/or other similar programs

29.5.5.2.4.3 Software Readiness Assessment

A software readiness assessment is an evaluation of software technical and management maturity at predefined points in the software lifecycle. The purpose of a software readiness assessment is to determine the readiness of the software for the next phase of development or for an upcoming designated event and the associated risk of proceeding. The ultimate goal is to provide objective, independent feedback to the program's software stakeholders. In performing a software readiness assessment, the software readiness assessment team evaluates risk by assessing software planning, technical performance, execution progress, and software quality, and their effect on software cost and schedule. To accomplish this, the software readiness assessment team applies the assessment criteria provided in [17] to three perspectives of the software development program as it evolves through the lifecycle: products, processes, and resources. While [16] was developed for space segment software, the assessment criteria in the document can easily be adapted to ground segment software because very few of the criteria are space segment-specific.

The objectives of the Software Readiness Assessment are shown in Table 29-25.

Table 29-25. Software Readiness Assessment Objectives

Principal Objectives	Major Review Products
a. Determine the readiness of the software for the next phase of development or for an upcoming designated event b. Determine the associated risk of proceeding with the next phase of development or designated event c. Assess software planning, technical performance, execution progress, and software quality, and their effect on software cost and schedule d. Provide objective, independent feedback to the program's software stakeholders	Contractor Products: Note: The products depend upon where in the life cycle the assessment is performed. See software-related products under the major technical reviews SFR/SDR (Table 29-4), SAR (Table 29-5), PDR (Table 29-6), CDR (Table 29-7), and the software-specific joint technical reviews SBPR, SBRAR, SBDR, SBTRR, SBER (Tables 29-9 through 29-13).

29.5.5.3 Independent Reviews Lessons Learned

The following lessons learned apply to the independent assessments:

- The contract should specify that the contractor will provide adequate corporate resources to support the independent review process.
- The contract should reference the charter, scope and needed resources (data, information, and personnel) to support the review process.
- It is advised that the contract provides a liaison between the contractor and review team.
- The contractor's documentation should be current.
- Roles and responsibilities should be clearly documented and understood by all stakeholders.
- The contract should provide office and/or conference space, management information system (MIS) support, and administration support as required.

The following lessons learned apply to the software-specific independent assessments:

- Software architecture evaluations, software process appraisals, and software readiness assessments should be specified in the contract statement of work or in a Section H, Special Provision clause.

- The contract should specify the expected schedule for these software-specific independent assessments and the frequency of their occurrence. In particular, the contract should specify the relationship of the timing of the software-specific independent assessments to ground segment increments and software builds.

- The contract should specify the type of team to be used (acquirer only, contractor only, or acquirer and contractor).

- Documents specifying the process to be followed and criteria to be used in the software-specific independent assessments should be specified as compliance or reference documents on the contract.

- Timely and effective response to deficiencies or issues identified by software independent assessments should be tied to award or incentive fees on the contract.

29.6 References

1. Adams, R. J., et al., *Software Development Standard for Mission Critical Systems*, TR-RS-2015-00012, The Aerospace Corporation, El Segundo, CA. March 17, 2014.

2. USAF SMC et al., *Interagency Cooperation for Mission Assurance*, Memorandum of Understanding Among Air Force Space and Missile Systems Center, National Reconnaissance Office, Missile Defense Agency, and National Aeronautics and Space Administration, February 2011.

3. Guarro, S. B., and W. F. Tosney, Editors, *Mission Assurance Guide*, TOR-2007(8546)-6018, Rev. B, The Aerospace Corporation, El Segundo, CA. August 1, 2012.

4. Defense Acquisition University (DAU) ACQuipedia, Performance Measurement Baseline (RMB), Primary Functional Area: Business https://dap.dau.mil/acquipedia/Pages/ArticleDetails.aspx?aid=be3618b1-abdb-41fd-8336-fd5c924c9917 2014.

5. IEEE, *Draft Standard for Technical Reviews and Audits on Defense Programs*, IEEE P15288.2/D5.2, September 2014.

6. Abelson, L. A., et al., *Integrating Software Topics into the Request for Proposal*, TOR-2011(8506)-117, The Aerospace Corporation, El Segundo, CA. July 19, 2012.

7. Department of Defense, *Work Breakdown Structure for Defense Materiel Items*, Department of Defense Standard Practice, MIL-STD-881C, 3 October 2011.

8. DAU, ACQuipedia Integrated Baseline Review, Primary Functional Area: Program Management. https://dap.dau.mil/acquipedia/Pages/ArticleDetails.aspx?aid=cf5eb839-0881-4044-9f23-2c675726b481#

9. Office of the Undersecretary of Defense (OUSD), *The Program Manager's Guide to the Integrated Baseline Review Process*, April 2003.

10. Peresztegy, L. B. and C. E. O'Connor, *Technical Reviews and Audits of Systems, Equipment, and Computer Software*, TOR-2007(8583)-6414, Rev. 1, Vol. 1, The Aerospace Corporation, El Segundo, CA. January 30, 2009.

11. Peresztegy, L. B. and C. E. O'Connor, *Technical Reviews and Audits of Systems, Equipment, and Computer Software*, TOR-2007(8583)-6414, Rev. 1, Vol. 2, The Aerospace Corporation, El Segundo, CA. January 30, 2009.

12. Department of Defense, *Operation of the Defense Acquisition System*, Department of Defense Instruction (DODI) 5000.02, January 7, 2015.

13. Unell, A. et al., *Evaluating Software Architectures in Space and Ground Systems*, ATR-2012(9010)-12, The Aerospace Corporation, El Segundo, CA. July 17, 2012.

14. Kazman, Rick, Mark Klein, and Paul Clements. *ATAM: Method for Architecture Evaluation*, Pittsburgh, PA: Carnegie Mellon University (CMU)/Software Engineering Institute (SEI), 2000.

15. CMMI Product Team, *CMMI® for Development, Version 1.3 (CMMI®-DEV, V1.3)*, Carnegie Mellon University (CMU)/Software Engineering Institute (SEI), CMU/SEI-2010-TR-033, November 2010.

16. CMMI Institute, *Standard CMMI® Appraisal Method for Process Improvement (SCAMPI℠), Version 1.3a: Method Definition Document for SCAMPI A, B, and C*, CMMI Institute-2013-HB-001, October 2013.

17. Eslinger, S., L. J. Holloway, and R. Wilkes, *Space Segment Software Readiness Assessment*, TOR-2011(8591)-20, The Aerospace Corporation, El Segundo, CA. June 3, 2011.

29.7 Bibliography

Technical Handbooks

Corporate Chief Engineering Office, *Independent Review Team Process, Version 1.0*, The Aerospace Corporation, Aerospace Technical Instruction –TI-RA-2.0, September 3, 2010.

Nelson, N., and G. Arnold, *Independent Review Process – Overview and Best Practices*, ATR-2009(9369)-20, The Aerospace Corporation, El Segundo, CA. August 20, 2009.

Bonesteele, R., et al., *Independent Program Assessment (IPA) Planning, Preparation, Execution, and Follow-Up (Rev 1)*, TOR-2010(8506)-4 Rev 1, The Aerospace Corporation, El Segundo, CA. March 17, 2011.

Guide Books

United States Air Force (USAF) Space and Missile Systems Center (SMC), *Independent Readiness Review Team*, SMC Guidebook SMC-G-1203, 2009.

USAF SMC, *Readiness Review Process*, SMC Guidebook SMC-G-1204, 2009.

Other References

Tosney, W., P. G. Cheng, and J. B. Juranek, *Guidelines for Space Systems Critical Gated Events*, TOR-2014(8583)-8545, The Aerospace Corporation, El Segundo, CA. May 9, 2008.

29.8 Acronyms

A	analysis
ADM	acquisition decision memorandum
AoA	analysis of alternatives
ASP	Acquisition Strategy Plan
ASR	alternate system review
ATAM	architecture tradeoff analysis method
ATP	authority to proceed
ATR	aerospace technical report
CARD	cost analysis requirements document

CDD	capability development document
CDR	critical design review
CDRL	contract data requirements list
CMMI®	Capability Maturity Model® Integration℠
CMMI®-DEV	Capability Maturity Model® for development
CMU	Carnegie Mellon University
CONOPS	concept of operations
COTS	commercial off-the-shelf
CPD	capability production document
CPI	cost performance index/continuous process improvements
CTE	critical technology elements
CTS	consent to ship review
D	demonstration
DAU	defense acquisition university
DBDD	database design document
DOD	Department of Defense
DODI	Department of Defense instruction
DR	discrepancy report
EMD	engineering and manufacturing development
EPOT	enhanced phase operations transition
ERR	enterprise readiness review
EVM	earned value management
FCA	functional capability audit
FD	full deployment
FDD	full deployment decision
FOC	full operational capability
FQT	formal qualification test
FRR	flight readiness review
GRA	government reference architecture
HB	handbook
HVAC	heating, ventilation, air conditioning
I	inspection
I&T	integration and test
IA	information assurance
IAS	information assurance standards
IBR	integrated baseline review
ICD	interface control document
IDD	interface design document
IEEE	institute for electrical and electronics engineers
IMP	integrated master plan
IMS	integrated master schedule
IOC	initial operational capability
IP	intellectual property
IPA	independent program assessment
IRRT	independent readiness review team

IRS	interface requirements specification
IRT	independent review team
ISP	information support plan
LCSP	lifecycle sustainment plan
LRR	launch readiness review
MIL	military
MIS	management information systems
MRR	mission readiness review
OAR	operational acceptance review
OCD	operational concept document
ORD	operational requirements document
ORR	operational readiness review
OT&E	operational test and evaluation
OUE	operation utility evaluation
OUSD	office of the undersecretary of defense
PCA	physical configuration audit
PDR	preliminary design review
PESHE	programmatic environmental safety and health evaluation
PMB	program management baseline
PMR	program management review
PO	program office
POE	program office estimate
PPP	program protection plan
PRR	production readiness review
PSA	pre-ship audit
PSR	pre-ship review
RA	risk assessment
RFC	request for change
RFP	request for proposal
SAR	software requirements and architecture review
SBDR	software build design review
SBER	software build exit review
SBPR	software build planning review
SBRAR	software build requirements and architecture review
SBTRR	software build test readiness review
SCAMPI℠	Standard CMMI® Appraisal Method for Process Improvement
SCP	spacecraft control processor
SCR	system/segment completion review
SDD	software design document
SDP	software development plan
SDR	system design review
SE&I	systems engineering and integration
SEI	software engineering institute
SEP	systems engineering plan
SETA	system engineering and technical assistance

SFR	system functional review
SFS	software product specification
SIQT	software item qualification testing
SLOC	source lines of code
SM	service mark
SMC	space and missile systems center
SMBP	software master build plan
SRR	system requirements review (or segment requirements review)
SRS	software requirements specification
SRT	systems requirement tests
SSAA	system security accreditation agreement
STAR	system threat assessment report
STD	standard
STD	software test description
STP	software test plan
STR	software test report
SUM	software user's manual
SVD	software version description
SW	software
SWAMP	software acquisition management plan
T	test
TDS	technology development strategy plan
TEMP	test and evaluation master plan
TES	test and evaluation strategy
TOR	technical operating report
TR	technical report
TRA	technology readiness assessment
TRD	technical requirements document
TRR	test readiness review
TT&C	telemetry, tracking, and commanding
USAF	United States Air Force
WBS	work breakdown structure

Chapter 30
Reliability, Maintainability, and Availability

Jya-Syin W. Chien, Roland J. Duphily, and Yum Tong Lee
Acquisition Risk and Reliability Engineering Department
Mission Assurance Subdivision

30.1 Introduction

Ground systems play important roles in supporting the success of a space mission. A typical ground system consists of a master control center, multiple receiving and monitoring stations, a multiple uplink antenna, and communication networks connecting all assets. The ground system is designed to provide services throughout the different phases of a space mission, which covers the time from space vehicle launch, normal operation of multiple space vehicles, anomaly analyses and resolution, and sometimes the decommission of a satellite service. As a result, the design of the ground system hardware is often complex and, most likely, has built-in redundancy, and its software would be composed of thousands of lines of codes and often developed over a long period.

A successful space mission starts with satellites launched with effective ground control and support. After being launched onto the planned orbit, the on-orbit satellites will need a high performance ground control system to achieve the designed level of performance and mission availability. The consequence of failure to implement effective reliability and maintainability programs ranges from the catastrophic failures of a program's space and/or ground assets, to degrading space system performance, shortening of satellite service life, delaying satellite launch schedule, and/or higher program life cycle and sustainment costs.

Guidance on reliability engineering practice published recently [1,2] mostly focus on the implementation on space systems and programs. Due to the distanced physical location of the satellites for any maintenance activities to be feasible, the reliability guidance for the space systems has limited content on maintainability and is mostly silent on the subject of using availability as one of the performance measures. This chapter focuses on the reliability, maintainability, and availability (RMA) tasks to be accomplished for a ground system over the program lifecycle to support the success of a space mission.

30.2 Definitions

Availability The degree to which a system or component is operational and accessible when required for use.

Maintainability The ability of an item to be retained or restored to, a specified condition when maintenance is performed by personnel having specified skill levels, using prescribed procedures and resources, at each prescribed level of maintenance [3].

Reliability The probability of an item to perform a required function under stated conditions for a specified period of time.

Utilization The average system operational usage measured in time over a specific time interval.

30.3 Acquisition Life Cycle

RMA activities start early during the concept studies phase of an acquisition when performance thresholds and objectives are defined. These activities continue and evolve as the acquisition progresses into mature phases, during which the performance thresholds and objectives need to be realized in system design and verified and demonstrated in the delivered products. Later on in the acquisition life cycle, test and operational experience need to be tracked and reflected in RMA estimates. To be a living RMA program, lessons learned need to be captured and wrapped into the operating phase of the project or into the organizational learning process [3].

30.3.1 Concept Studies Phase

During the concept studies phase, RMA capability requirements are established in support of the overall system performance thresholds or objectives. Some of the requirements will be quantitative, such as system-level RMA requirements. Some of the requirements, however, will be qualitative. A requirement for no single point of failures (NSPOFs,) for example, is one. Reliability engineers will also apply various factors and considerations into the trade analyses during this phase to explore alternative concepts and technological approaches.

Some high-level mission-specific RMA features are also identified at this stage. Reliability, availability, and maintainability figures of merit at various operational phases and conditions are standard sets of RMA performance measures. Sometimes other RMA features to be tracked are also identified at this stage. For example, if the continuous functioning of the ground operation over a given interval under a specific operating condition is a required feature of the acquisition, then continuity or dependability would be specified as one of the system RMA requirements.

30.3.2 Concept Design Phase

In the concept design phase, RMA requirements continue to mature and are allocated down to sub-system levels. During the concept design phase, feasibility studies take place via trade studies, and RMA results are some of the key inputs to the life cycle cost estimates, some redundant architecture becoming necessary for the system to achieve its qualitative and quantitative RMA requirements.

Source selection and contract award take place during this phase. Key RMA activities during this phase include evaluating and selecting contractors that have experience and who can execute the reliability and maintainability program plans successfully. Because space programs and ground programs are quite different in their technical nature, past experience in one does not necessarily translate into the successful execution in the other.

Once past contract award, the customer needs to provide the prime and subcontractors with sufficient support in picking up the role of RMA execution. The customer reliability team will take the responsibility of contract compliance monitoring and technical supporting.

30.3.3 Preliminary Design Phase

The RMA activities during the early stage of the preliminary design phase include allocation of requirements down to lower-level system designs and checking their consistency against required operational capabilities. As the baseline design architecture becomes stable, reliability and maintainability (R&M) requirements can be verified incrementally, and risk and problems identified and resolved timely and effectively. Activities of the preliminary design phase culminate in the program preliminary design review (PDR) milestone

Compared to the space systems, ground systems supporting space mission are often more software intensive and have a complex communication network design; they also tend to have more external interfaces. The identification of single points of failures in both software and hardware systems, as well as in the interfaces, as well as their resolution, often poses a challenge for ground systems during the preliminary design phase.

30.3.4 Detailed Design Phase

At the detailed design phase, a baseline design has reached its maturity with supporting analyses to be reviewed at the critical design review (CDR). RMA predictions based on best available data are compared against the system-level RMA requirements. The RMA activities at this stage should include the

implementation and management of a failure reporting and corrective action process to ensure timely identification and resolution of failures/anomalies during testing in the build and operations phase.

The extended external interfaces and complex communication network of the ground systems have introduced new challenges in the cybersecurity, information assurance (IA), and security areas. Potential interference from the implemental IA system and external noise has to be addressed in the detailed design phase, as well the software RMA consideration.

30.3.5 Build and Operations Phase

System qualification, integration, and acceptance tests are planned as the system is built, integrated, and fielded. Failure data and operating anomalies, as well as real maintenance data, are reported, analyzed, and collected. Reliability engineers apply statistical methods to these system-specific data to assess actual system RMA requirements and to identify and implement corrective actions or improvements as necessary. The emphasis of RMA activities during this stage is also on controlling processes and expanding the failure reporting and corrective action recording process.

Note that the intensive use of software in the ground control system makes the software RMA a subject that has attracted a lot of attention in recent years. Some acquisition strategies adopt a design assurance approach and elect not to specify quantitative RMA merits for software, relying instead on design assurance processes defined in the Software Development Plan to make sure that the as-designed software meets the performance requirements. Reference of that subject can be found in the RTCA guidelines [4].

An effective R&M program supports all the major acquisition activities through the full system life cycle.

30.4 Reliability Program for the Ground Segment

The fundamental difference in RMA programs between a space program and a ground support program is that, once launched, the space system would have less opportunity to perform corrective and preventive maintenance work, and so more emphasis is put on high quality reliability programs. In addition, as a given ground segment is most likely to be designed to support multiple launches and on-orbit operations, availability of the system is one of the performance parameters used for resource planning. A typical reliability program for a ground segment includes availability as one of the performance metrics by applying maintenance data collected from the activities planned in the maintainability program.

30.4.1 Reliability Program Management, Surveillance, and Control

To successfully meet all reliability objectives, the contractor needs to develop an effective reliability organization and a comprehensive reliability program plan. The reliability program plan lays out the approaches and methodologies needed to meet all specified system reliability requirements. It also defines a design process by which reliability-related risks are mitigated and all reliability requirements are properly allocated down to the lower-level systems and eventually reflected in the final design and verified. The plan also needs to address the issue of software reliability and the interaction with the software development plan. Finally, the reliability plan defines timely documentation methods and reporting systems that facilitate the control of the reliability program.

Failure reporting, analysis, and corrective action system (FRACAS) is the closed loop failure analysis and corrective action system established to ensure that all failures are documented, analyzed for root cause, and that timely corrective actions are taken to reduce or prevent recurrence. The FRACAS serves as a management tool to identify, correct, and prevent further recurrence of all failures occurring in hardware and software during fabrication, system debugging, qualification tests, engineering test, acceptance test, flight tests, and on-orbit failures.

The customer needs to be an active participant in the contractor and subcontractor failure review board (FRB) process to ensure that the root cause of a failure or an anomaly is adequately identified and prevented from occurring in the future. The failure analysis and corrective action results need to be well documented and easily retrievable for use in future on-orbit failure investigations.

Most contractors have an automated FRACAS database, which includes their subcontractor failures. Preliminary copies of each failure report are typically contractually deliverable, submitted within a week after the failure occurs. Data packages also may be submitted by the contractor to the customer for review prior to the FRB. Summaries of open failure report status generally are presented at regular contractor program reviews. In addition, delivered hardware typically includes completed test failure reports as part of its end-item data packages.

30.4.2 Reliability Design and Development

30.4.2.1 Ground System Reliability Metrics

Figures of merit, such as mean time to failure (MTTF), probability of failure, reliability, mean time to restore function (MTTRF), and so on, provide guidance

to the design team as an indication of the performance durability of the system. Figures of merit also help determining the necessary part quality, redundancy, and part stress levels needed to meet the expected mission success criteria. Through analytical and empirical methods, the intended uses, mission profile, success criteria, and environments of the system are translated into realistic system-level reliability performance parameters to be specified in system development specifications and requirements documents. It is important that the reliability performance requirements are specified in terms of operating conditions and that they are properly allocated down to lower subsystems, for verifying compliance of the reliability requirements.

Ground segment RMA metrics depend upon the duration and the frequency of the occurrence of a specific planned or unscheduled event occurring to the system or to certain equipment of the system under a given operational condition. Factors considered for the RMA metrics include operational conditions, effects of operators, maintenance personnel, management decision, and physical stresses arising from the events involved. As the ground command and control systems support a range of missions including space launch, space surveillance, communications, and navigation, the duration and frequency of the support could be 24-hour-7-day down to a fraction of a day, a week, or even a month, depending on the nature of the supporting functions. RMA metrics and the related RMA specifications are more meaningful when defined with respect to the specific time period of interest. For example, launch mission reliability mostly is defined for periods when scheduled maintenance activities are not allowed, while operational availability will include both scheduled and unscheduled outages. Most of the definitions in this section are based on the material in reference [5].

A ground system RMA program puts more emphasis on quality corrective and preventive maintenance, and the ground system RMA performance. Consequently, the ground system RMA metrics are measures that reflect the reliability performance of the system and the effectiveness of the maintenance program.

Typical measures used for the RMA metrics and the use of standard RMA definitions has a better chance to eliminate confusion and to promote consistency in reporting operational incidences and performance. Basic terminology, and a common set of RMA metrics definition are provided in the following sections to standardize these terms and elements.

30.4.2.2 Terms Used on Time and Event Elements

30.4.2.2.1 Time Elements

Time is the principal variable in the quantification of RMA metrics. There are multiple categories of time (e.g., the possessed and non-possessed times, and the up and down times).

Some ground systems may be setup for operations only for a short period of time and RMA metrics may be explicitly defined to measure the performance during this specific time period. This partial time period of interest may be designated by a specific operational interval name for clarity purpose (e.g., launch interval for the Space-lift Range System). This specified, operational time interval replaces the possessed time in all metric definitions if the resultant metrics are applied only to this specific time interval.

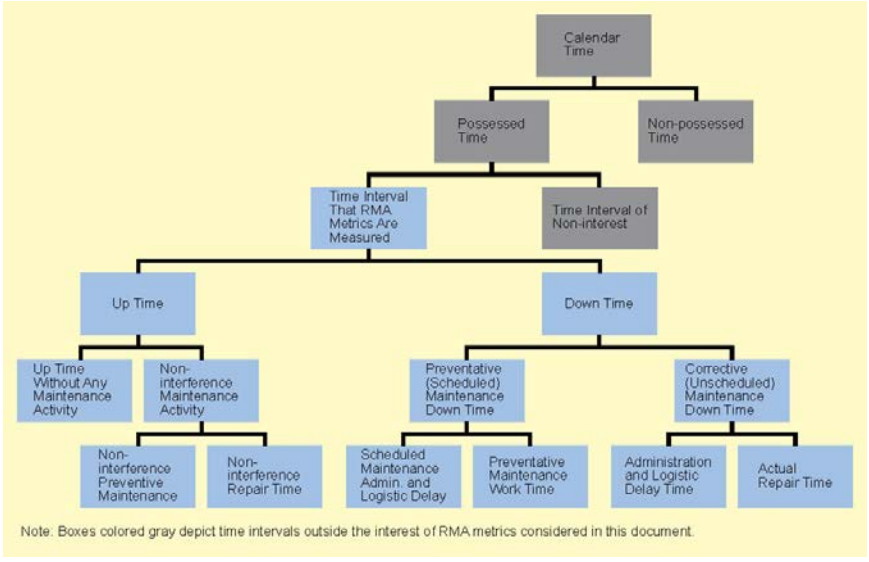


Figure 30-1. Hierarchy time elements relationship [5].

Figure 30-1 illustrates the relationships among time elements. Most of these time elements are equal to the sum of the time elements depicted immediately below it. The exceptions are time elements associated with the maintenance activities (or simply called maintenance times). Time intervals of maintenance events performed on different parts of the system may overlap with the maintenance work time.

The following is a listing of the common terms related to time and their definitions:

Possessed time is the time of possession of the system or equipment. It represents the time period that the system or equipment is in its operational environment.

Non-possessed time is the period of time that the possession or the control of the system or the equipment is temporary turned over to a different organization for major modifications or upgrades. If the non-possessed time is relatively short compared to the possessed time, then the possessed time is often approximated by the clock time.

Specified operational interval time is a specified time period of possessed time during which RMA performance of the system or equipment are measured. Metrics applicable to the specified interval are based on the events which occur during this interval of time.

Mission time is the operating time for some defined activities collectively known as "mission."

Down time is the time during which the system or equipment is down or is in a down state. Down time is the duration of a downing event. It is the portion of possessed time required to restore or maintain a system or an equipment item in mission or functionally capable condition. Down time may be scheduled or unscheduled.

Up time is the time during which the system or equipment is up or is in an up state. It is the time the system or equipment is capable of initiating or continuing a specified mission or capable of performing all its designated functions.

Maintenance time is the elapse time required to perform all maintenance actions associated with a maintenance event. Note that maintenance time does not include any administrative or logistics delay times. Maintenance times may be categorized into preventive and corrective maintenance time according to the types of maintenance events. Maintenance times may also be separated into functionally interfering maintenance times (of either corrective or preventive in nature) and non-interfering maintenance times. It may be desirable to keep track of the different categories of maintenance events.

Repair time is the corrective maintenance action time required to return a specific failed part of a system or a failed equipment item to operational status. Repair actions that are postponed to a later date are considered as repairs and not scheduled maintenance. Repair time is the time spent to repair the specific failed part and it is not necessary equal to the elapsed time to restore the failed part

functions which could be achieved by a part replacement. The repair time is usually not determined at the system level because system function failures may not be repaired immediately, but are often restored quickly by substitution with a spare unit. The time to restore function is determined at the system level. Repair time excludes administrative and logistics delay times.

Time To Restore Function (TTRF) is the down time, or the clock time, to reinstate a failed function of a system or subsystem function to operable condition. It is a measure of the function down time which is not necessarily equal to the repair time. It is a measure of how fast a system or subsystem may recover to an up state from a down state due to a critical failure. TTRF includes administrative and logistics delay times.

Administrative and logistics delay time (ALDT) is a period of down time during which no repair or maintenance is performed due to delays in administrative processing, parts delivery, assignments of maintenance personnel or equipment, or transportation. A system usually has an insignificant administrative and logistics delay time for the scheduled preventive maintenance. The primary reason for computing ALDT is to assess the time delay to recover from an unanticipated downing event. Therefore, ALDT is determined based on the delay times associated with failure events.

30.4.2.2.2 Event Elements

In the following we provide some definitions on event terms that are critical for the understanding and communication of the RMA metrics.

Critical Failures are system degradation, indications of failures, or actual failures that prevent a system from being used to perform a specified mission or function.

Downing Events are events in which the system is unable to initiate or continue its mission or functions. Downing events include all critical failures plus any scheduled events, such as preventive maintenance or training, which prevent the system from performing its designated operation(s). An equipment item is considered down if it is temporarily disabled due to failure or maintenance of other equipment. Major modifications and upgrades are not downing events because these events occur outside of the time interval of interest by the metric definitions and their times constitute non-possessed time.

Failure is the occurrence of physical degradation or a logic fault. A failure at the equipment level where there is no redundancy resulting in the inability of the equipment to perform any one or more of its design functions. A ground system usually has redundancy and a failure of its subsystems may not cause the loss of any designated system function. Therefore, equipment failure is usually a critical

failure of the equipment, but a failure of a system is not necessary a critical failure of the system. Preventive maintenance and maintenance training that may disable a system or a piece of equipment are not failures.

Maintenance Event is one or more preventive maintenance or repair actions to preserve or to restore the equipment to working order. Maintenance actions include: (1) troubleshooting due to any type of failure or malfunction; (2) preventive maintenance; (3) repair; and (4) servicing. Maintenance events can include maintenance actions performed while the equipment is in an operational state. Thus, maintenance events may include more events than downing events. However, installation and test for major modifications or upgrades are not considered maintenance events; these events occur outside of the time interval of interest by the metric definitions and their times constitute non-possessed time. Preventive maintenance includes the routine servicing or replacement of equipment prescribed on a calendar, hours of operation or usage basis, and the restoring of non-failure degradation. An individual ground system may have its own unique reporting of maintenance events which would affect the maintenance event count and the times of these events. This suggests that metrics dependent on maintenance events are not as robust as metrics that depend on downing events. The effects are more noticeable at the system level than the lower hierarchical equipment level. A maintenance event at the equipment level is less ambiguous and may be correlated to an equipment downing event.

30.4.2.2.3 RMA Metrics

Mission Success Rate (MSR) is the ratio of successful missions to the total number of missions, i.e.:

$$MSR = \frac{\sum Successful\ Missions}{Number\ of\ Missions} \quad (1)$$

Mean Down Time (MDT) is the average duration of downing events; i.e.:

$$MDT = \frac{\sum Down\ Time}{Number\ of\ Downing\ Events} \quad (2)$$

Mean Time Between Downing Events (MTBDE) is the average up time between downing events; i.e.**:**

$$MTBDE = \frac{\sum Up\ Time}{Number\ of\ Downing\ Events} \quad (3)$$

Mean Time Between Critical Failures (MTBCF) is the average up time between critical failures**:**

$$MTBCF = \frac{\sum Up\ Time}{Number\ of\ Critical\ Failures} \quad (4)$$

Mean Time Between Failures (MTBF) is the average up time between failures. It is usually determined for the equipment level only when its function relative to the specific mission has not been defined:

$$MTBF = \frac{\sum Up\ Time}{Number\ of\ Failures} \quad (5)$$

Mean Time To Restore Function (MTTRF) is the average time to restore a function after a critical failure:

$$MTTRF = \frac{\sum Time\ To\ Restore\ Function}{Number\ of\ Critical\ Failures} \quad (6)$$

Mean Repair Time (MRT) is the average repair time:

$$MRT = \frac{\sum Repair\ Time}{Number\ of\ Failures} \quad (7)$$

System or Equipment Reliability (R(t)) is the probability that a system or equipment will operate for a time period equal to *t* without any critical failures. Under certain specific conditions, this term can be approximated by an exponential distribution:

$$R(t) = e^{-\left(\frac{t}{MTBCF}\right)} \quad (8)$$

Note: Equipment usually has no redundancy which implies that an equipment failure usually is a critical failure of the equipment and that equipment MTBCF is equal to equipment MTBF.

Mission Reliability ($R_m(t_m)$ or MR) is the probability that the mission is successful up to time t_m, starting from the beginning of the mission. If the time function of mission success can be approximated by an exponential distribution, the equation for MR will be presented as:

$$R_m(t_m) = e^{-\left(\frac{t_m}{MTBCF}\right)} \quad (9)$$

Operational Availability (A_o) is the fraction of time that a system in an operable state for its intended operational environment:

$$A_o = \frac{MTBDE}{MTBDE+MDT} \quad (10)$$

Operational Dependability *(D_o)* is the fraction of time that a system in an operable state for its intended operational environment, given that the system down times were due to activities on scheduled preventive maintenance.

$$D_o = \frac{MTBCF}{MTBCF+MTTFR} \tag{11}$$

Note: For a continuous operating system without scheduled downing preventive maintenance as part of its normal operation, D_o is identical to A_o.

Mean Mission Duration (MMD) is the average duration of a mission:

$$MMD = \frac{\sum Mission\ Time}{Number\ of\ Missions} \tag{12}$$

Note: There are more than one MMD definitions used by the RMA community. The one listed here is typically used by the ground system RMA analysts.

Administrative and Logistics Delay Time (ALDT), as defined in the previous section, is a period of down time during which no repair or maintenance is performed due to delays in administrative processing, parts delivery, assignments of maintenance personnel or equipment, or transportation**:**

$$ALDT = \frac{\sum^{All\ failures} Administrative\ and\ Logistics\ Delay\ Times}{Number\ of\ Failures} \tag{13}$$

Mean Maintenance Time (MMT) is the average maintenance time per maintenance event**:**

$$MMT = \frac{\sum Maintenance\ Time}{Number\ of\ Maintenance\ Events} \tag{14}$$

Mean Maintenance Man-Hours (MMMH) is the average maintenance man-hours per maintenance event:

$$MMMH = \frac{\sum Maintenance\ Man\_Hours}{Number\ of\ Maintenance\ Events} \tag{15}$$

Mean Time Between Maintenance (MTBM) is the average up time between maintenance events both scheduled and unscheduled. MTBM is usually computed at the equipment level. Maintenance events may be subdivided into different categories, and different MTBMs may be defined for the separate categories of maintenance events**:**

$$MTBM = \frac{\sum Up\ Time}{Number\ of\ Maintenance\ Events} \tag{16}$$

Turnaround Time (TT) is the average time associated with the preparation and configuration of the ground system for a back-to-back mission. The preparation and configuration of the back-to-back mission is defined as the time in-between the start of the mission and the end of the preceding mission. A back-to-back mission is one which has no provision for idle time prior to the second mission. Turnaround time is, generally, not considered to be a RMA term but rather a performance requirement that may affect the RMA performance:

$$TT = \frac{\sum Preparation\ \&\ Configuration\ Time}{Number\ of\ Back\_to\_Back\ Missions} \tag{17}$$

Utilization is the average system operational usage measured in time over a specific time interval. It is expressed as a percentage of the possessed time. Utilization is not a measure of the RMA performance but a performance requirement that may affect the RMA metrics**:**

$$Utilization = \frac{\sum Operational\ Usage\ Time}{Possessed\ Time} \times 100 \tag{18}$$

30.4.2.3 Reliability Models and Predictions

For a ground system that supports complex architectures with single or multiple elements of space, launch, and ground segments, it is imperative to develop a high-level system of systems reliability and/or availability requirements, and to flow these high-level requirements down to appropriate lower elements. It is a living model that evolves with the design and assists with making decisions during trade studies of how to operate various element combinations and meet overall mission success probabilities.

Reliability block diagrams (RBDs) graphically represent the hardware and software needed for success, operating duty cycles, redundancy types, and any available work-around. When comparing competing designs, quantification of RBDs helps to determine which design concept is the most reliable or has the lowest probability of failure. Results of these analyses are contractually deliverable and part of design review packages.

During the design and development phases, probabilistic reliability models and failure data sources need to be independently reviewed for adequacy. RBDs or fault trees used to model the system also are reviewed, as are failure rates deemed reasonable for active and standby conditions are reviewed, Numerical results are also reviewed for reasonableness when compared to similar system.

30.4.2.4 Failure Modes and Effects Analysis and Failure Modes, Effects, and Criticality Analysis

The Failure Modes and Effects Analysis (FMEA) or Failure Modes, Effects, and Criticality Analysis (FMECA) process effective tools in the decision making process, provided it is a timely iterative activity. A FMECA is identical to an FMEA except that, in addition to the consideration of each hardware element, its failure modes, and the effects on higher levels, one also evaluates the associated criticality of its consequence. The FMEA/FMECA should start at a high level as early as preliminary design and architecture information is available. It will then be extended to lower levels as more detailed information become available. Implementing FMEA or FMECA late in the program, or restricting its application, may dramatically limit its use as an effective tool for improving the design or improving the design process.

When any design or process changes are made, the FMEA/FMECA is updated and the effects of new failure modes introduced by the changes are carefully assessed. Although the FMEA/FMECA is primarily a reliability tool, it provides information and support to safety, availability, maintainability, logistics, test, and maintenance planning, and the design of fault management, also called failure detection, isolation, and recovery (FDIR) design. The use of FMEA/FMECA results by several disciplines assures consistency and avoids the proliferation of requirements and the duplication of effort within the same program. Results of these analyses are likely to be contractually deliverable with periodic updates and summarized with critical design review (CDR) packages.

An important benefit of the FMEA/FMECA process, if independently reviewed and evaluated, is the identification and control of credible single-point failures. Because the ground system often has multiple interfaces with external systems, the FMEA/FMECA should also include the investigation of the interfaces to identify the single point failures resulting from interface design and to ensure that none of these single point failures propagate leading to undesirable consequences.

30.4.2.5 Critical or Limited Life Item Control

One of the reliability program products is the mission and safety critical items list. Items whose failure would directly affect system or personnel safety, mission success, or operational readiness are often included in the mission and safety critical items list for control and tracking. The early identification, tracking, and control of critical items through the preparation, implementation, and maintenance of a critical items list is a valuable risk prevention strategy and will provide valuable inputs to a design, development, and production program. From the critical items list activity, critical design features, tests, inspection points, and procedures can be identified and implemented that will minimize the

probability of failure of a mission or loss of life. Results of these analyses are typically contractually deliverable with periodic updates and part of design review packages.

30.4.2.6 Worst Case and Parts Stress Analysis

A worst-case analysis is performed to ensure that the system is adequately designed for its expected operating conditions. The most sensitive design parameters are analyzed, including those subject to variations that could degrade performance. The adequacy of design margins in electronic circuits, optics, electro-mechanical and mechanical items are demonstrated by analyses, test, or both. The analyses consider all parameters set at worst-case limits and worst-case environmental stresses. Part parameter values for analyses include manufacturing, temperature and cumulative aging mechanisms (e.g., stress, soldering, moisture, shock) updated with design changes. The analysis results are presented at various design reviews.

Stress analysis is performed on components that are newly designed or modified according to the system needs. This includes the design modifications incorporated into a commercial off-the-shelf/non-development item (COTS/NDI), which often lacks design details that might bring problems at latter stages of the program. The stress analysis is often conducted using predicted worst-case environmental and load conditions. An alternative approach to avoid failure caused by over-stressing the elements in the system is to implement a sophisticated de-rating policy in the early design.

30.4.2.7 Parts Reliability Analysis

Electrical, electronic, optical, and mechanical part failure rates are the basic building blocks of probabilistic reliability predictions. Therefore, confidence in the predictions is very much dependent on having accurate failure rates derived from credible sources or test data with appropriate adjustments for quality, end-use environment, stress levels, and temperature levels. MIL-HDBK-217F [6] is a source for obtaining reasonable failure rate estimates based on a consistent and uniform methodology when failure data is insufficient. Since the last release of MIL-HDBK-217F was more than a decade ago, there is an urgent need for an updated common failure rate and reliability model repository [7]. For new parts (e.g., hetero-junction bipolar transistors, field programmable gate arrays [FPGAs], application-specific integrated circuits [ASIC] etc.) it is important that the part qualification process be independently reviewed by a team consisting of domain experts, quality assurance, and reliability engineers to validate the design [8].

30.4.3 Reliability Verification and Evaluation

During the design and development phases, the reliability and availability predictions are compared against reliability and availability requirements. To verify that the actual system reflects the reliability and availability design, various tests are conducted during the build and early operating phases of the project. To help validate reliability predictions, an independent evaluation can be performed at the part quality level, via accelerated part life tests , against de-rating criteria and/or stressed conditions, and at various system-level tests as described in [9].

30.4.3.1 Accelerated Life Testing

The contractor establishes and maintains an accelerated life testing (ALT) program to detect and correct any inherent design and manufacturing flaws and to determine product robustness of mission critical items. Selection criteria are established to identify ALT candidates. Criteria and candidates are made available for technical review. ALT is used during development in an iterative fashion beginning at lower level of assembly and progressing to higher levels of assembly until sufficient margins have been verified. Test methods include a series of individual and combined stresses applied in steps of increasing intensity (well beyond the expected field environments) until failure or a malfunction is obtained. Failure modes are analyzed for root cause and corrective action.

30.4.3.2 Environmental Stress Screening

An effective environmental stress screening (ESS) program is created and maintained so that workmanship failure can be identified early and removed from equipment. The program includes development of ESS profiles based on the analysis of the operating conditions. Temperature and humidity are often among the items of great concerns. Other factors such as ground and lighting events are also on the list for consideration. To determine the most effective screening profiles, the ESS program includes feedback to latent and undetected failure modes, and workmanship defects into the FRACAS. ALT results may also be used as a baseline for determining initial ESS considerations.

30.5 Maintainability Program for Ground Systems Supporting Space Missions

Compared to the RMA programs for the space vehicle, maintainability plays a much more important role for ground systems as the sustainment is long term and resource intensive [10–12]. Over the years, more practitioners have realized the need for the maintainability programs to be “reliability-centered,” or to

prioritize the maintenance tasks according to their impacts on mission reliability performance [13].

30.5.1 Maintainability Program

For a ground system, there are opportunities for preventive and corrective maintenance activities, depending on the nature of the support it provides in the space mission. The objective of the maintainability program is to assure that the maintainability requirements of an acquisition are attained to serve the needs of the mission. The benefits include improving operational readiness, effectively managing maintenance resources, reducing maintenance manpower needs, reducing life cycle cost, and recording and providing maintenance data essential for management. After the award of the contract, the contractor will establish and maintain an effective maintainability program that is planned, integrated, and developed in conjunction with other design, development, and production functions with the objective of the most cost-effective overall program management.

In a ground system maintainability program, it lists the management and technical resources for the program execution, plans, process, procedures, schedule, and controls for the maintenance activities to assure that the maintainability requirements will be met. These procedures will be established with the understanding that maintenance is an integral part of the design process. Maintenance procedures to be developed will have the design process in mind and include fault detection and corrections at the organization, engineering, and depot levels. Interfaces with other programs, and analyses such as reliability and availability analyses, logistics support analyses, and risk analyses and management, are also key factors in developing the maintenance program plan. In addition, the maintenance program is to be established consistent with the criticality of the missions, the strictness of the requirements, the complexity of the architecture and design, the innovation of the technology, the experience of the industry on a similar system, and the maturity of the fabrication and manufacturing techniques required.

30.5.2 Quantitative Maintainability Requirements

Quantitative maintainability requirements for the system and equipment are included in the system and lower-level specifications. Figures of merit used in maintainability requirements are often specified in terms of functions of time, man-hours, or in attributes of fault detection and isolation. Examples of quantitative measures to be considered for contractually specified requirements include mean-time-to-repair (MTTR), mean-time-to-restore-system (MTTRS), MTTRF, probability of fault detection, and so on. These and other RMA terms and metrics are defined and discussed in Section 3.2.2 of this chapter. Cost-

effectiveness, traceability, mission operational values are also some of the considerations when selecting maintenance measures and figures of merit.

A maintainability analysis will consider system failure mode and its diagnosis and corrective maintenance tasks. It also considers the skill levels and number of people required, support equipment and tools required, and technical orders to be prepared for the maintenance tasks. It will also include the identification of the preventive maintenance tasks with the objective of extending the service life.

30.5.3 Maintainability Verification, Demonstration, and Evaluation

The maintainability test plan describes the verification, demonstration, and evaluation of the maintainability requirements. The plan is to be prepared by the contractor and approved by the procuring authority.

For a complex system, maintainability verification can be challenging due to a strenuous operating environment preventing collection of reliable maintenance data. Quite often maintainability data are obtained from a laboratory environment and controlled failures seeded for recording of time for diagnosis and repair duration, as well as other controlled data. Laboratory data and field data are then combined in support of maintainability requirement and process verification, demonstration, and evaluation. The approach of combining laboratory data and field data has to be carefully examined so that the conclusions reached will realistically reflect the maintenance practices during operation.

30.6 References

1. Ingram-Cotton, John B., Myron J. Hecht, Roland J. Duphily. *Reliability Program Requirements for Space Systems,* TR-RS-2007-00013 (SMC-S-013), The Aerospace Corporation, El Segundo, CA. July 10, 2007.

2. Englehart, William C. *Space Vehicle Systems Engineering Handbook*, TOR-2006(8506)-4494, The Aerospace Corporation, El Segundo, CA. November 30, 2005. Restricted distribution.

3. SMC-G-002, *Reliability and Maintainability,* Space and Missile Systems Center Standard, Air Force Space Command. June 13, 2008.

4. RTCA/DO-278, *Guidelines for Communication, Navigation, Surveillance, and Air Traffic Management (CNS/ATM) Systems Software Integrity Assurance*, Washington, DC. RTCA, Inc. March 5, 2002.

5. Lee, Yum Tong. *Reliability, Maintainability and Availability Metrics for Air Force Ground Systems Supporting Satellite Operations,* TOR-96(1565)-1, The Aerospace Corporation, El Segundo, CA. August 1996.

6. MIL-HDBK-217F, *Reliability Prediction of Electronic Equipment*, Military Standardization Handbook, Department of Defense, January 2, 1990.

7. Kawamoto, Jack T. *Evaluation of Existing Failure Rate (FR) Handbooks and Contractor FR Models,* TOR-2013(3009)-3, The Aerospace Corporation, El Segundo, CA. December 3, 2013.

8. RTCA/DO-254, *Design Assurance Guidance for Airborne Electronic Hardware*, Washington, D.C., RTCA, Inc., April 19, 2000.

9. MIL-STD-1833, *Test Requirements for Ground Equipment and Associated Computer Software Supporting Space Vehicles*, Military Standard, Department of Defense, November 13, 1989.

10. MIL-STD-470B, *Maintainability Program for Systems and Equipment*, Military Standard, Department of Defense, May 30, 1989.

11. MIL-STD-471A, *Maintainability Verification, Demonstration, and Evaluation*, Military Standard, Department of Defense, March 27, 1973.

12. MIL-HDBK-472, *Maintainability Prediction*, Military Standardization Handbook, Department of Defense, May 24, 1966.

13. *DOD Guide for Achieving Reliability, Availability, and Maintainability*, Department of Defense, United State of America, August 3, 2005.

30.7 Acronyms

AF	Air Force
ALDT	administrative and logistics delay time
ALT	accelerated life testing
Ao	operational availability
ASIC	application-specific integrated circuits
CDR	critical design review
COTS	commercial off-the-shelf
Do	operational dependability
DOD	Department of Defense
ESS	environmental stress testing
FDIR	failure detection, isolation, and recovery

FMEA	failure modes and effects analysis
FMECA	failure modes, effects, and criticality analysis
FPGA	field programmable gate arrays
FRACAS	failure reporting, analysis, and corrective action system
FRB	failure review board
IA	information assurance
MDT	mean downtime
MMD	mean mission duration
MMMH	mean maintenance man hours
MMT	mean maintenance time
MR	mission reliability
MRT	mean repair time
MSR	mission success rate
MTBCF	mean time between critical failures
MTBDE	mean time between downing events
MTBM	mean time between maintenance
MTTF	mean time to failure
MTTR	mean time to repair
MTTRF	mean time to restore function
MTTRS	mean time to restore system
NDI	non-development item
NSPOF	no single points of failure
PDR	preliminary design review
R&M	reliability and maintainability
R(t)	system or equipment reliability
RBD	reliability block diagrams
RMA	reliability, maintainability, and availability
SMC	Space and Missile Systems Center
TT	turnaround time
TTRF	time to restore function

Chapter 31
Software Reliability

Myron J. Hecht
Software Acquisition and Modeling Department
Software Engineering Subdivision

31.1 Introduction

Ground systems are software-intensive systems with the following characteristics: (a) engineering and development content of such systems is overwhelmingly software, (b) the execution platforms are general purpose information technology-grade hardware, operating systems, and middleware (e.g., application servers and enterprise service buses), (c) significant portions of the system consist of modified, previously developed software (often referred to as "software reuse"), (d) system architectures are based on networks and incorporate redundancy in both hardware (HW) and software (SW) components. Because of the dominant role of software in both the composition of ground systems and the systems they control, ground systems reliability is primarily determined by the reliability of the software.

Achieving operationally suitable reliability and availability in software intensive systems is challenging. The DOD Reliability, Availability, and Maintainability (RAM) Guide [1] analysis indicated that 85 percent of software intensive projects finished over schedule or budget; half of projects doubled original cost estimates; projects slipped an average of 36 months; and one-third of projects were canceled. Software reliability is a product of good identification of software requirements in the systems engineering process, robust software design, thorough analysis, and robust testing. Inadequate software reliability can double or triple field support and maintenance costs [1].

Software has a dual nature. Its static nature is as a set of instructions. In that state, software does not fail and the concept of software reliability is irrelevant. However, when it is executed, software can and does cause failures. Because software failures occur only during execution of the software, and because execution of the software requires processors, memory, and network hardware, it is more accurate to think of such failures are "software-involved failures" or "software-caused failures." Extending this concept to software reliability, we can define this as the probability that an item or system under evaluation will not fail in a defined time interval due to a software involved failure.

A misconception of software failures is that they are all deterministic. That is, if a specific set of inputs that once caused a failure, it would always cause a failure. It is true that some software failures are deterministic, but many are not.

This was shown as early as 1985 in tandem systems where 90% of the failures could be resolved by restart [2, 3] and more recently in (1) space systems where more than 30% of the reported failures were not deterministic [4] and (2) a command and control system where 28% of the reported failures were not deterministic. [5] The reason is that many software failures are caused by unique environmental circumstances such as timing, event sequences, or deadlock. For example, in the case of a memory leak, the same set of inputs would sometimes cause a failure but sometimes would not. Deterministic failures are often referred to as "Bohrbugs," whereas random failures are referred to as "Heisenbugs" or "Mandelbugs."

Software intensive system reliability, availability, and maintainability (RAM) requirements encompass quantitative and qualitative attributes of the design as well as the system and software development process. Quantitative RAM requirements are generally stated at the system level in the specification associated with the request for proposal (RFP) (e.g., system mean time between failures [MTBF], mean time to restore, and availability [MTTR]). In the past, RAM has been interpreted by the acquisition authority as applying only to hardware, and hence the actual total system failure rate (i.e., including software) was far higher—often orders of magnitude higher—than what was predicted by the development organization. As a result, systems were significantly delayed in deployment as failures and problems in software that resulted in less-than-satisfactory operational suitability, safety, or effectiveness were resolved leading to the consequences reported by the DOD RAM guide [1].

31.2 Definitions

Acquisition organization the organization acquiring the system (e.g., the Air Force Space and Missile Center, see also, "development organization").

Availability the degree to which a system or component is operational and accessible when required for use. [6] Note: Often expressed as a probability.

Contractor a type of development organization which forms a legal contract to develop a system to specific requirements with an acquisition organization.

Development organization the organization developing the system for the acquisition organization (usually a contractor, see also, "acquisition organization").

Failure mode the physical or functional manifestation of a failure [6]. A system in failure mode may be characterized by slow operation, incorrect outputs, or complete termination of execution.

Failure the termination of the ability of an item to perform a required function [7]: Note: in software, a failure is an event that can be caused by the interaction of the data, timing, or event sequences with a fault.

Fault a defect or flaw in a software or hardware item or component [7]. Note: In software, faults exist in requirements, design, code, and data. Not all software faults cause failures, and a single software fault can cause many failures. Whereas failures are events, faults can be represented as states. Faults can result from errors occurring during development or production or they can be the consequence of a preceding failure. For example, a preceding hardware or software failure might cause a buffer to fill thereby resulting in the fault of a full buffer, which subsequently leads to another software failure caused by the full buffer.

Government an acquisition organization that is a component of a U.S. Federal or State government which forms a contract with a development organization to create a system that meets specific requirements.

Maintainability the ability of an item to be retained in, or restored to, a specified condition when maintenance is performed by personnel having specified skill levels, using prescribed procedures and resources, at each prescribed level of maintenance and repair [8]. When used in a mathematical sense, the probability that an item can be restored to operation within a specified time or a derived quantity thereof (e.g., average or mean time to restore).

Reliability the probability of an item to perform a required function under stated conditions for a specified period of time [1].

Reliability growth the improvement in reliability that results from correction of faults. [6] Note: for the purposes of this document, “faults” and “reliability” growth refer to software.

Software reliability capability of the software product to maintain a specified level of performance when used under specified conditions [6]. Note: Often expressed as a probability.

31.3 Objectives

The objectives of the software reliability program are to:

- Address software reliability throughout the lifecycle beginning in the early conceptual stages of system design.

- Utilize holistic design methods to address hardware, software, and human factors elements of system reliability—not as compartmentalized concerns, but via comprehensive integrated approaches.

- Address failure detection and recovery provisions through a complete analysis of the software architecture and design.

- Develop a test and evaluation process including models, testing, and measurement to continuously assess the reliability, availability, and maintainability of the entire system including both the hardware and software.

As noted by a recent National Research Council report, software-intensive systems and subsystems merit special reliability scrutiny throughout the lifecycle beginning in the early conceptual stages of system design through operation [9]. The practices described in the following section support this finding.

31.4 Practices

Practices for achieving high levels of reliability and availability in ground systems are embodied within a RAM program plan. The authority of the reliability program plan should come from the System Engineering Plan. The results should feed into the information assurance as well as into the system engineering activities.

Figure 31-8 shows the key program elements and data flows of a RAM program for software intensive systems such as ground systems. In accordance with DoDI 5000.02 [10], the program is either a component of, or expands upon, the systems engineering plan (in the case of the acquisition organization) or systems engineering management plan (for the development organization). The foundation of the RAM program is the RAM Program Plan shown in the center of the top row of Figure 31-1.

Requirements definition is the formulation of the top-level RAM requirements for the system. Allocation is the process by which these top-level requirements are decomposed into lower-level elements. Then, the modeling and prediction activity occurs in which a more detailed model is created and used, together with failure rate, recovery probability, and restoration time data, to calculate the expected system reliability. It differs from the allocations in that the allocations are top-down, and the predictions are bottom-up. After (or while) the requirements are being defined, the system architecture and design are being created. Because the most significant aspects of the design and architecture are

related to software in ground systems, these are software intensive activities. As part of this process, the reliability requirements affect the use of redundancy, recovery times, interfaces, component interactions, and coding practices. These are embodied in the design for reliability block.

The reliability modeling, analysis, assessment, and allocation tasks provide the probabilistic failure assessments whose parameters are, in part, based on the results of reliability testing and the Failure Reporting and Corrective Action System (FRACAS) program. Outputs of this modeling will feed the security (information assurance), system safety, and system engineering processes. The Failure Modes and Effects Analysis (FMEA) task predicts the qualitative and (partial) quantitative failure behavior of the system. These predictions are corroborated and modified by the observed behavior as recorded in the reliability test and FRACAS activities.

The RAM program activities shown in Figure 31-1 span multiple program phases. Table 31-1 shows the relationship between the development phases as defined in DoDI 5000.02 [10] and the RAM program.

Table 31-1. RAM Activities and Program Phases

Development Phase [10]	RAM Program Plan	RAM Contractual Provisions	Requirements Definition and Allocation	RAM Modeling	FMEA/CA	Design and Milestone Reviews	FRACAS	Reliability Growth	Reliability Testing
Pre-Milestone A	X	X	X						
Post-Milestone A	X	X	X	X	X	X			
Post-Milestone B			X	X	X	X	X	X	
Post-Milestone C				X	X	X	X	X	X

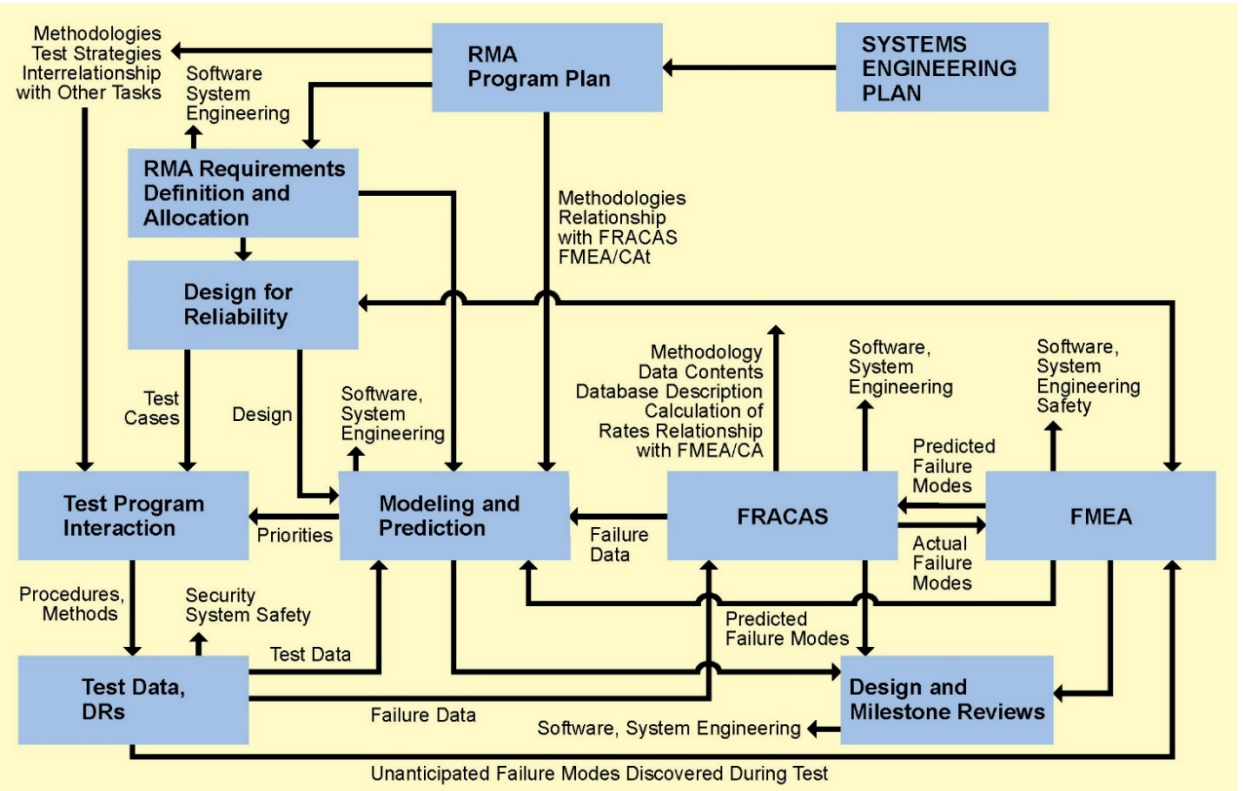


Figure 31-1. RAM program key elements and data flows for software intensive systems.

31.4.1 RAM Program Plan

DoDI 5000.02 requires that "the Program Manager … formulate a comprehensive R&M program using an appropriate strategy to ensure reliability and maintainability requirements are achieved" and that the "… program [be] an integral part of the systems engineering process" [10]. Separate RAM programs should be defined for both the acquisition organization and for the development organization. The RAM program of the acquisition organization will be focused on requirements definition, oversight, and reporting. The acquisition organization's RAM program plan may be separate or a part of the overall system engineering activity and an element of the overall system engineering plan (SEP). The development organization should formulate a program similar to that shown in Figure 31-8 to comply with DoDI 5000.02 instructions. Specific requirements for the plan should be documented in the contractual provisions. These provisions—in both the contract and in the RFP that precede it—are necessary to achieving the reliability requirements and must be drafted with care.

When complete and approved by the acquisition organization, the development organization's RAM plan should be incorporated either in its entirety or by reference into the SEMP so that it becomes a compliance document for the ground system development program.

31.4.2 Requirements Definition

The importance of RAM requirements have been stated by both the Under Secretary of Defense of Acquisition, Technology and Logistics (USD[AT&L]) DoDI 5000.02, and the Joint Chiefs of Staff through the CJCSI 3170.01H, "Joint Capabilities Integration and Development System (JCIDS) Instruction" [11]. DoDI 5000.02 mandates a Reliability, Availability, Maintainability and Cost (RAM-C) Report which requires the definition of such requirements the time of Concept Development [12]. CJCSI 3170.H and its supporting implementation manual [13] provide guidance on the development of key performance parameters (KPPs) and key system attributes (KSAs). They set forth two reliability-related KPPs, operational and materiel availability, and two reliability-related KSAs, materiel reliability and operations and sustainment (O&S) cost. Values for these KPPs and KSAs are defined in the Capabilities Description Document (CDD), carried forward into the system technical requirements document (TRD), and allocated to lower level elements through program requirements and design documents.

While the CDD and associated KPPs and KSAs, as well as the TRD, are the province of the government, lower-level requirements (for elements, subsystems, CIs, etc.) are in the domain of the development organization. Requirements for reliability, availability, and restoration time from the TRD are

usually not defined with adequate specificity for verification by analysis or testing, and hence, the acquisition organization must ensure that lower-level RAM requirements are written so that they are verifiable.

Like all other requirements, RAM requirements at all levels must be written with care. Table 31-3 lists common requirements, definition problems, and solutions for software-intensive systems.

Table 31-3. Software-Intensive Requirements Problems and Solutions

Requirement Definition Problem	Solution
Not clearly stating that software failures are covered by reliability requirements: Requirements must explicitly state that software failures are included in the scope of system availability and reliability requirements (nearly all functional requirements are implemented in software).	Define system failures to include software, and require reliability verification methods to account for integrated hardware and software functionality.
Not accounting for multiple modes and states: A software failure might result in a loss of a communication channel, a security function, or the entire telemetry tracking and control (TT&C) function. In addition, there may be several system-defined states (e.g., full capability, degraded, etc.).	Define such functions and states unambiguously so that failures in the component hardware and software elements can be correctly allocated or "scored" during testing.
Not clearly defining countable failures: During testing external events may cause system outages (e.g., HVAC or power failures). Requirements need to be extremely clear as to what is a countable or relevant failure under these circumstances.	Formulate requirements so that when external failures induce system anomalies, it is possible to determine whether these induced anomalies are countable as failures.
Not defining outage durations: Recovery times affect operational impact. Software-intensive systems are recoverable from some classes of failures (hang, crash, delayed response) using either a restart or a switchover to a standby unit. If the recovery time is short, there may be little or no effect on the operation or mission.	Specify quantitative response time and recovery time requirements for each function (can be defined in terms of classes)
Not distinguishing between failures and operational positions versus failures of services or functions. Failures at operational positions may or may not have the same impact as loss of the central function.	Specify reliability and availability (including response time and recovery times) for both operational positions and central services on which they depend.
Not defining the interval over which operating time is measured. Operating time measurements affect reliability, availability, and restoration time. Not defining the interval of these measurements results in ambiguous verification criteria.	Define operating time intervals (e.g., "availability over any 30 day interval," "restoration time measured over a period of a minimum of 1 year").

Requirement Definition Problem	Solution
Not accounting for diagnostics, logistics, and administrative delays. Availability should be defined to account for both intrinsic and extrinsic factors in restoration.	Define terms such as "inherent availability," "operational availability," "achieved availability."

Because software (and system) reliability is assessed to the extent which requirements are being met, validation of the correctness, consistency, and completeness of the requirements is critical. The following methods and processes have proven to be quite valuable in the past, and should be used as a part of a requirements validation effort:

- *Modeling and simulation:* Modeling and simulation should be used to set and evaluate performance parameters requirements affecting software (accuracy, response time, throughput, numeric precision, sampling frequency), quality communications of services requirements (bit error rate, dropped packets, latency, availability) in the data transport layers, requirements for responses to failures and anomalous conditions, and human/software or system interactions. The models utilized in the requirements phase can be maintained and enhanced for the purposes of analysis and evaluation of subsequent phases of the software development (architecture, design, and implementation) as well.

- *Use of relevant "golden rules" and "lessons learned":* Golden rules or lessons learned are excellent sources of requirements and should be reviewed as part of the requirements validation process. In general, the information contained in these items was generated as a result of a significant mishap that either occurred or was avoided. As such, the experience is quite valuable

- *Results of FMEAs and Fault Trees:* Fault trees are top-down analyses and FMEAs are bottom-up. Both of these analyses evaluate how failures can occur and what mitigations are required. The results of these analyses should be compared to existing requirements to identify missing or incomplete requirements.

31.4.3 RAM Allocations

The purposes of RAM allocations are to:

- Assess the feasibility of meeting operational availability and materiel availability quantitative requirements for both hardware and software in the ground system.

- Establish lower level quantitative requirements for subsystems, software, and components of the ground system.

Allocations are usually performed analytically using reliability block diagrams of both hardware and software elements. Known software failure rate and recovery time data should be used if available.

The following is the procedure for performing the allocation:

1. Define the system hierarchy in accordance with the architecture starting at the top level down to the lowest allocable elements (executable software and hardware LRUs)

2. Allocate unavailability top down through each level of the hierarchy and rebalance if necessary. Account for redundancy if present

3. Define restoration times for each of the elements

4. Calculate mean time between failures (MTBFs) based on allocated unavailabilities and restoration times

The procedure may be modified depending on available data and other circumstances. The absence of data is a problem, especially for software, because the reliability of software components cannot be credibly estimated in advance of execution in the actual operational environment. In such circumstances, estimates based on other relevant past experience can be used. For example, commercial (COTS) or widely used open source software (OSS) (e.g., operating systems, DBMSs, web servers, application servers, enterprise service buses, etc.) can be assumed to have failure rates of 0.0001 per hour. With such allocations for the COTS or OSS software, allocations for the developed software can be developed. Credible failure rate allocations for developed software should not be less than 0.001 per hour (corresponding to a 1000 hour MTBF) unless the executable modules are small or development measures for high integrity systems (i.e., safety critical or cybersecurity critical) are being used.

31.4.4 Software Architecture, Design, and Implementation for Reliability

There are some software-related technical issues that should be addressed in developing ground intensive systems. As noted in the software reliability appendix of the DOD RAM Guide, “The highest payoff efforts for reliability and maintainability for both hardware and software is in the front-end [1].” Although the discussion concentrates on software architecture, design, and implementation, hardware issues should not be ignored. The National Research

Council has stated that "Design methods should be pursued to address hardware, software, and human factors elements of system reliability—not as compartmentalized concerns, but via integrated approaches that comprehensively address potential interaction failure modes" [9].

31.4.4.1 Architecture definition

Software architectures are developed with a variety of considerations, and a complete discussion of the subject is well beyond the scope of this chapter. On the other hand, general treatments of software architectures frequently downplay reliability, availability, and sustainment concerns in favor of other issues. Therefore, the issues identified in this subsection should be integrated with those identified in more generalized guides, books, or papers:

- *Extent of redundancy:* A key top-level design question is the extent of redundancy needed for the mission. The most conservative approach is stated as fail operational/fail operational/fail-safe. In this configuration, five redundant computers are required. If one processor fails, normal operations are still maintained. Two failures result in a fail-safe situation, because the three remaining processors allow for a majority vote. However, cost and schedule considerations precluded such a strategy.

- *Extent of modularity:* Modularity is generally considered a desirable architectural attribute because it facilitates uncoupled development, integration of revised components, and utilization of previously developed (COTS) components. However, a consequence of increasing software modularization is an increase in the number of interfaces. These interfaces need to be maintained, and may in themselves introduce increased complexity and delays that may increase the likelihood of a failure. Defining the optimal balance between modularization and integration is a tradeoff that is architecture and implementation-specific. However, safety considerations should factor in this decision.

- *Point-to-point vs. common communications infrastructure:* Even if the hardware communications structure utilizes common bus or shared network communications, inter-software process communications can either be point-to-point or utilize a common software communications mechanism (often called an object broker or an enterprise service bus for object-oriented and service-oriented architectures [SOAs]). The use of a common software communications bus has advantages in reducing the interdependencies among software elements and use of common inter-process communications constructs (message structures,

protocols, ports, and remote process calls or method invocations). On the other hand, a message bus introduces vulnerabilities in terms of lost or delayed messages, message integrity, or other failure conditions. From the safety perspective, an optimal communication architecture may incorporate elements of both point-to-point and common software communication mechanisms.

- *Extent of Abstraction:* Abstraction layers such as hypervisors and virtual machines or SOAs add intermediate software components ("stacks") between applications. The advantages they provide (flexibility, extensibility) must be weighed against their disadvantages (quality of service impact, presence of bottlenecks and single points of failure). The design tradeoffs are beyond the scope of this chapter, but those involved with mission assurance should ensure that they are being properly addressed.

- *Redundancy and Diversity:* Redundancy and diversity are key elements for increasing the safety of the software architecture. Redundancy means replicated servers and executable processes, and increases reliability in the presence of random failures. Diversity is the use of an alternate (even if not optimal) means of fulfilling a requirement in the event that the primary mechanism fails and increases reliability in the presence of design failures and malicious actions.

31.4.4.2 Software design

The following are some of the issues that need to be considered by developers of the software design with respect to reliability:

- *Traceability:* Requirements should be traceable to the functional elements (procedures, functions) or classes (depending on methodology) defined in the design. For functional oriented architectures, duplications in functions should be minimized and where there is duplication in the functional decomposition, the rationale should be explained. Decomposition of higher-level functions into lower-level functions should be complete (i.e., no aspects of the higher level functionality should not be mapped to a lower level function). As explained in the following text, object-oriented architectures sometimes complicate this traceability because functionality may be distributed across multiple classes.

- *Exception handling and other failure behaviors:* Exception handlers should consider all failure conditions defined in the requirements and identified by means of safety analyses (fault hazard analysis [FHAs],

FMEAs, etc.) at lower levels of the architecture. Exception handlers should also consider all of the failure conditions likely to occur within the module (or class) itself based on an analysis of the design and prospective implementation. Where possible, exceptions should be handled as close to the locations in the code where they are generated (i.e., as soon as possible). The design should not allow exceptions to propagate without a documented rationale.

- *Diagnostics capabilities:* a related concern is the design of the software to meet allocated requirements from higher levels as well as additional requirements imposed by the architecture to sense and report on failures in its environment (even if the application itself is not in a degraded condition). Special attention should be paid to response time anomalies, priority inversion, and resource contention. The diagnostic capability of the system as a whole will largely depend on the diagnostic capabilities in all of the constituent software components.

- *Redundancy management:* The redundancy management constructs in the design should be totally consistent with those defined in the architecture. Further information on top-level considerations in the design of software implemented redundancy management schemes occurs later in this chapter discussion of fault tolerance.

- *Implementation language:* The implementation language and runtime environment (including virtual machines) should be capable of realizing the design. Of particular concern are language features that support exception handling, timing constraints, checkpointing and logging, and recovery.

- *Interfaces:* Interfaces among software modules should be completely defined and include not only arguments for the inputs and outputs of the function or object itself, but also additional parameters for status, error handling, and recovery. Interfaces should be designed "defensively", i.e., to minimize failure propagation (parameter validation prior to use, strong typing, exception handling when constraints are violated).

- *Class library definition and inheritance:* For object-oriented architectures the definition of base and derived classes should be consistent and traceable to both the requirements and the architecture. This is sometimes more complicated than in functionally oriented languages because the functionality necessary to meet a specific functional requirement may be distributed across several classes.

- *Compatibility with hardware and resource constraints:* The software allocated to each hardware element should conform to memory, processor capacity, and interface constraints.

- *COTS and Non-developmental runtime elements:* Existing software components and runtime elements (operating system, protocol stack, data base management system, messaging middleware, runtime libraries, etc.) should be configuration controlled, well characterized (as to resource requirements, safety, failure behavior) with respect to the intended use, and documented.

31.4.4.3 Coding and Implementation

Many reliability concerns are common with other concerns related to software quality, readability, and maintainability and are not repeated here. The following are specific concerns related to software reliability that apply during the coding phase:

- *Selection of subroutine or class libraries, and runtime environments:* The runtime libraries and other environmental components that support the developed software should conform to the constraints of the architecture and design and should provide the necessary capabilities to support desired failure behavior—including:

  - Reliability, performance, throughput

  - Failure response, detection and recovery (e.g., whether execution should be sustained on unaffected threads or tasks if a failure or detectable degradation occurs in another thread)

  - Diagnostics requirements

- *Definition of suitable coding standards and conventions:* Modern languages support many constructs such as dynamic binding, tasking, dynamic memory reclamation ("garbage collection"), and other features that increase power but also make runtime behavior extremely difficult to predict. Failures can occur through resource exhaustion (e.g., memory leaks), priority inversion, buffer overflows, and deadlock. The decision on coding practices is project- and application-specific, Coding standards and conventions can enhance safety by considering such issues as

  - Policies on dynamic memory allocation in safety critical systems (generally, it should not be allowed)

- "Defensive" coding practices for out-of-range inputs and response times

- Exception handler implementation

- Coding to enhance testability and readability

- Documentation to support verification

- Interrupt versus deterministic timing loop processing for safety critical software

- Policies on allowable interprocess communications mechanisms (e.g., point-to-point vs. publish and subscribe)

- Permitted use of dynamic binding (an alternative is static "case statements")

- Policies on initialization of variables (some standards prohibit assignment of dummy values to variables upon initialization in order to enable detection of assignment errors in subsequent execution)

- Use of "friend" (C++) or "child" (Ada) declarations to enable testing and evaluation of encapsulated data code during development without requiring the subsequent removal of "scaffold code".

- For object-oriented languages, limitations on levels of inheritance in order to prevent "accidental inheritance" due to introduction of variables with the same name or variable misspelling.

- *Coding tools, static analysis tools, and development environments:* Coding tools, static analyzers, and integrated development environments can be used for many purposes including automated documentation (both internal and external) generation, enforcement of coding standards, debugging, diagnosis of potentially troublesome coding practices (not necessarily covered by coding standards), cross reference listing, execution profiles, dependency analysis, design traceability, and many other purposes. Organizational software development process definitions should describe the use of these tools to reduce the likelihood of defect introduction and increase the likelihood of their removal once discovered.

- *Automated Code Generation Tools:* Newer techniques based on object-oriented design or model-based development have resulted in tools that can go directly from design to executable code. These techniques are likely to become of great importance for real time control systems. Among the advantages of such tools is the ability to generate an executable design that can be evaluated prior to detailed coding. In a variety of tool suites the tests of the design can be used as tests of the auto-generated software. Since safety is a main concern, it is important to note that the design tests should be very robust with respect to detection and managing anomalies and faults.

31.4.5 System Modeling and Prediction

The purposes of RAM modeling and prediction are to

- Demonstrate compliance with quantitative requirements listed above
- Investigate sensitivities of system RAM to subsystem and component failure rates, recovery times, and recovery probabilities
- Assess the conformance with reliability/availability growth plans and predictions

As noted in the introduction, software does not fail on its own, and it causes system failures. Thus, reliability modeling and prediction is a system-level task that includes both hardware and software.

The reliability model enable prediction of system reliability based on the underlying component failure rates, restoration times, and recovery probabilities. Because ground systems are software intensive with a large number of reconfigurable processing elements, traditional reliability block diagrams cannot adequately model the large state space. Therefore, modeling prediction should be performed using a discrete event simulation with Markov modeling as an adjunct for specific situations (e.g., tradeoffs of coverage vs. MTBF for an application cluster).

Much of the software in a ground system is COTS or OSS for which reliability and related performance parameters are known or for which measurements in a representative environment can be made from previous operational test and evaluation (OT&E) testing or based on past experience. Examples include the underlying operating system, messaging “middleware” layers, network monitoring and control software, and database management systems. Parameters estimated from empirical data and test data can support such estimates. Established methods are available to calculate point estimates and confidence intervals from reliability test and operational data [14] [15].

31.4.6 Failure Modes and Effects Analysis/Criticality Analysis

The FMEA is one of the most important, but labor intensive activities to ensure ground system software and system reliability. The purpose of the FMEA is to assess the response of the ground system to single, credible failures in any part of the developed, COTS, or government off-the-shelf (GOTS) software, whether in the infrastructure, core services, or mission services layers. The FMEA will identify failure modes, effects, severities, detection capabilities, and mitigations (either automatic or manual recovery actions). The FMEA is be the basis for assessing conformance with the following requirements:

- Single point of failure
- Failure propagation
- Detection for diagnostics and repair

The FMEA analysis should also identify critical items in both hardware and software whose failure can have a major or catastrophic effect on the mission. For software, these components should be either eliminated from the design (often impossible) or subject to special coding practices, test, and verification methods.

FMEAs should be performed on both system hardware and software using predefined methods and worksheets based on the requirements of a defined a program-approved methodology. Many such methodologies have been documented in a variety of industries that include both hardware and software [14]. If sufficient, credible, quantitative failure rate information for a significant proportion of system hardware and software components are available, then a criticality analysis (CA) will be performed as well.

31.4.7 Failure Reporting and Corrective Action System

FRACAS tracks system failure events (which, in a ground system, are primarily software) from their occurrence to their resolution and subsequent analysis. The objectives of the FRACAS is to identify and correct:

- recurring root causes and trends (groups of failure events with common cause or failure mode),
- components with excessive failures and the associated development process problems,
- installation or operation procedures that can cause failures,
- requirements errors and omissions,
- unanticipated failure modes or severities and unsuccessful detection methods, and recovery actions (be fed back into the FMEA),
- design errors and omissions,

- deficient testing or other quality checks that fail to prevent defective items from being integrated or deployed, and
- time trends of occurrence for the overall system and lower-level categories (including components) to provide early indications of reliability and availability growth or decline.

The FRACAS tracks events from all program activities and data should be collected from multiple sources including:

- The software discrepancy reporting (DR) tracking system (note that some but not all items in the DR tracking system represent failures; some represent requirements, design, coding, and documentation defects that are identified through review or analysis rather than execution),

- Developmental and operational testing and evaluation (see next section),

- Incident and maintenance reports from first-level maintainers of a related existing system (if the ground system under development is an incremental upgrade or extension of the existing system and the existing system has components common with the new system),

- Data from COTS hardware and software suppliers whose components are included in the system under development and where such failure data will have an impact on design or operation, and

- The FRACAS systems of any subcontractors involved in system development. The subcontractor FRACAS systems should be specified to be compatible with the development organization.

The data included in the reports should be sufficient to meet the objectives outlined above and should include:

- Unique failure identification number
- Time and date of failure
- Computer (or other hardware element) on which failure occurred
- Software item failing (if a software failure)
- Failure title (summary)
- Description of failure (describing symptoms)
- System on which failure occurred
- Release or increment
- Activity being executed at the time of the failure
- Detection method (how was failure found)
- Restoration method (immediately after failure)

- Failure/outage duration
- Chronology/history of the resolution (opened, screened, assigned, fixed, verified)
- Cause
- Development stage at which the discrepancy was inserted
- Development stage at which the defect was removed
- Corrective actions or disposition
- Severity

Where possible, the data items should use a fixed set of standardized entries. Thus, for example, the hardware and software items involve should come from a list defined from the configuration managed list of hardware and software components; the system on which the failure occurred should come from the system engineering plan; the release or increment should come from the software development plan; the cause should be taken from a program standard list of cause codes; the development stage from the software and systems development plan descriptions, and the corrective actions from the RAM program plan. However, there should always be another option, "other," which would allow the user to specify non-standard entries. Capturing unanticipated failures and other events is the primary value of the FRACAS.

31.4.8 RAM Growth Tracking

Reliability growth tracking was mandated by Congress in the Weapons Systems Acquisition Reform Act (WSARA) of 2009 [15]. DoDI 5000.02 [8] carries this mandate forward. Ground systems usually incorporate commercial, open source, and previously developed software and hardware whose reliability is not under the control of the development organization and therefore is unlikely to grow (with the exception of configuration and installation errors related to system integration). Thus, reliability growth is most likely to occur in project-developed software. On the other hand, maintainability and availability growth can be achieved by reduction of restoration time (a measure of maintainability) which is in turn most influenced by system monitoring, prognostics (predictive indicators of conditions likely to lead to failures), and diagnostics (indicators of items that failed for which corrective action or replacement is necessary). Such metrics can be affected by the manner in which all items (not only developed software) are integrated, monitored, and installed in the ground system.

31.4.8.1 Software Reliability Growth

The assessment of software reliability growth is based on the assessment of failure data over time. Data can be used during software development to measure time between events, analyze the improvement resulting from removing errors and making decisions about when to release or update a software product version.

Unfortunately, software reliability data gathered during development is not representative of the system's operational reliability because (a) the software is not representative of the operational configuration, (b) the testing input does not represent the typical operational profile, and (c) computer systems are operated sporadically rather than continuously. However, software discrepancy reports (DRs) can be used as a surrogate measure to qualitatively assess trends of reliability over development time if:

- Operational DRs are clearly tagged to distinguish other discrepancy reports related to documentation, requirements, or configuration errors or development and test networks

- DRs include information on the component in which the failure occurred and the time of the failure.

- Records of test time are maintained in a form that enables of normalization to a quantity related to operational time (i.e., DRs per test hour). Assessment of trends using merely calendar time will lead to flawed conclusions, because test intensity can vary significantly. This can be most easily seen by assessing the DR generation rate during the week of July 4 and the last two weeks of December.

- DRs include a symptom classification so that the degree of resolution of the underlying causes can be assessed. For example, a cluster of incorrect response failures might be related to an error in algorithm definition; a cluster of failures might be related to software configuration or integration errors; a cluster of late responses might be attributed to overloaded resources. A reliability growth can be derived by assessing the degree to which DRs with such clusters symptoms appear

Given credible time-trends of DR data, it is possible to project software reliability growth. A number of models have been proposed. One such family of models is based on non-homogeneous Poisson Processes (where the interarrival rate between failures changes over time) ranging from single parameter "learning curve" growth models to dual parameter Weibull models [16] to even more complex models with 3 or 4 parameters [17]. Other models are based on Bayesian approaches in which the predictions of future growth are conditioned on previous performance [17].

Scrutiny is necessary when assessing development organization claims of reliability growth using developmental DR data. While the DR generation rate (normalized for test or execution time) might show a decreasing time trend, it might be due to improvements in the software relative to the test plans and procedures (e.g., re-running of the same scenarios or input data) rather than to

inputs that are more operationally representative. In other words, a trend of reliability growth or a prediction of the number of "defects left in the code" may be more a prediction of the those defects remaining in the code that can be found through the testing program of the development organization rather than to the actual reliability of the software.

The extent of failure detection and recovery coverage can be used as a secondary measurement of reliability growth [8]. Such coverage reduces the likelihood of events that disrupt normal processing can propagate to the point that they become failures. For example, if a timeout is detected, the recovery provision might call for a default value to be substituted to enable continued processing to meet a hard response time requirement, thereby preventing a failure to meet that requirement. A software crash might be mitigated by switchover to a redundant copy (if, as noted above, the cause of the crash is a confluence of random external events rather than a deterministic cause, such as an incorrect formula that could cause an overflow in both versions of software). In some cases, small increases in coverage can have dramatic increases in time between failures. Figure 31-2 shows the results of a calculation with a 1700 hour MTBF and a 10-hour MTTR [18]. The graph shows that an increase of coverage from 90% to 95% results in an increase in the MTBF from 17000 hours to 40000 hours (the ordinate is graduated in 10,000 hour increments).

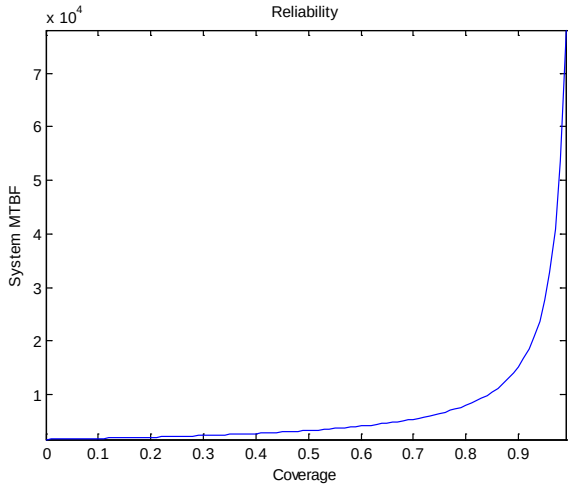


Figure 31-2. Impact of coverage on reliability (measured as MTBF).

Measurement of coverage increases requires the ability to track both symptoms and the components from which the failures originate.

Measurement of actual reliability growth can occur once system and software development is substantially complete and it is being operated. It is feasible to collect such data in ground system test and integration laboratories or at actual field sites because the computing platforms typically support network monitoring and failure logging. Reliability growth predictions and tracking can be facilitated by the methods described in MIL HDBK 189C [16] or the Software Reliability Engineering Handbook [17], using software tools such as SMERFS [19] developed by W. Farr and O. Smith of the Dahlgren Naval Surface Warfare Center and CASRE [20]. Although easy to use, the cautions on the data input to the models and limitations on the predictions of such software should be well understood before either the acquisition or development organization uses the results for major programmatic decisions.

31.4.8.2 Maintainability and Availability Growth

While not directly identified in the WSARA or DoDI 5000.02 [10, 15], maintainability and availability growth are as important in ground systems as reliability growth because ground systems are frequently operated continuously rather than for fixed mission times. It is not sufficient—a ground system which fails less frequently but has long duration outages as a result of the failures it does experience can be less operational than a system that fails more frequently but suffers only a brief interruption after each failure.

Furthermore, availability is actually a composite of reliability and restoration time. As either reliability increases or restoration time decreases, availability increases (informally known as “uptime”). It is often much easier to reduce restoration times than it is to increase MTBF to get the same benefit. For example, if a ground system has an achieved availability of 99% with a software mean time between failures (MTBF) of 100 hours and an average restoration time of 1 hour, but a required availability is 99.5% (which appear like a small difference but actually means a reduction in downtime of 50%), it might be much easier to reduce the average restoration time to 30 minutes by means of improved diagnostics and system monitoring than it would be to increase the software reliability from 100 to 200 hours (a 100% increase).

The FRACAS system can be used to track time trends in the discovery and resolution of failures. Ideally, as the system moves to the later stages of integration and testing, the rate of occurrence of failures should decrease and the times between such incidents should increase. This is referred to as “reliability growth” in DoDI 5000.02, which requires not only reliability growth tracking but also planning for reliability growth by program phase [21].

31.4.9 Testing, Evaluation and Verification

The DOD RAM Guide states that complex software intensive programs can spend more than half their total program effort on test and evaluation activities [1]. To make testing effective, planning must start during design. Testing should be done as early as possible and continue through the design process. A good testing program identifies problems early and when the cost of rework and fixes is lower [1].

The objective of a requirements-verification strategy is to develop a set of tests, analyses, inspections, and demonstrations that will show conformance to requirements within the schedule constraints of the program. The following are some of the issues and concerns that should be addressed in development of the verification strategy

- *Early planning:* Although much of the requirements verification work will occur at the end of the development phase, planning should start at the time of requirements definition so that adequate time is available for long lead-time items (e.g., scheduling of scarce resources such as simulation test beds, communications equipment, or even engine test stands for integrated hardware/software testing); acquisition of additional test tools and data collection devices can occur in a timely manner; and long duration analyses can be performed without causing a system delay. Moreover, development of the test program at the time of requirements definition can allow for efficiencies by allowing common verification methods and procedures for multiple requirements.

- *Verification methods for low observable parameters:* While many requirements can be verified by easily observed or acquired data, others generate more subtle data that may require additional technologies that must be acquired or developed by the project office. Examples of the latter include throughput, response time, testability, or safety.

- *Anticipating ephemeral failure behaviors:* The verification strategy should anticipate failure behaviors and plan for how this information can be captured—particularly if they are ephemeral and non-reproducible. For example, the tester's response to a system crash might be to ignore it, restart the system, and resume testing under the assumption that it is unlikely to recur. The verification program should develop methods to capture such events. While a mere nuisance to progress in the verification and test program on an individual level, collectively, the occurrence of such events could be indicative of certain trends and can help in identifying source of instability in the system.

- *Testing of diagnostics and failure isolation capabilities:* Diagnostic and fault isolation capabilities are of particular importance in space vehicles because of the limited visibility and access that experts have to the failed article. Inadequate diagnostics and fault isolation can be a direct cause of mission failure and can endanger crew lives on long duration missions. Unfortunately, it is one of the less glamorous aspects of the system development and is therefore quite vulnerable to downgrading in priority relative to other higher-visibility functions. The verification program should ensure that adequate coverage is given to testing and diagnostic and fault capabilities.

- *Capturing of unanticipated failures or behaviors:* The verification program should ensure that test plans and procedures have provisions for recording of unanticipated behaviors. While not necessarily relevant to the verification of requirements, they might be quite relevant to the opposite, and may be indicative of unanticipated reliability or safety problems.

- *Conformance to performance constraints:* Such constraints include throughput, and response time. Performance modeling may be possible. However, even without analytical or simulation models, simple addition of estimates of average processing and capacity requirements, as well as latencies for each step in the processing string, should be performed to ensure that at the very least the architectural can feasibly meet the requirements.

Verification of quantitative and qualitative RAM requirements for software intensive systems requires testing—particularly at the integrated system level during design test and evaluation (DT&E) and OT&E—because software reliability cannot be predicted with the same credibility as electronic hardware reliability. Thus, DT&E and OT&E should collect data and to assess:

- Failure rates
- Restoration times
- Switchover success probabilities

Using data from DT&E and OT&E for verification of system-level RAM requirements often necessitates that these quantities be decomposed from the system level to components whose failure rates and restoration times can be measured during testing. Methods set forth by the National Institute of Science and Technology can be used to calculate point estimates and confidence intervals for these quantities at either the component or system levels [22].

Because it is usually infeasible to dedicate large software systems for the duration of testing needed to assess conformance with RAM requirements, the testing program should exploit the operational experience during other testing activities to collect data necessary to verify the requirements. Using such data requires advance planning and coordination. For example, it is necessary to collect not only failure data, but also operating time and downtime. Furthermore, not all interruptions of operation are classified as failures. Some may be due to the nature of the functional testing that may be occurring, others may be due to the test environment. Thus, interruption data must be classified (a common synonym for classification used in testing is "scoring") after each testing event with relevant classification criteria agreed upon before the test readiness review (TRR) of each formal test.

Ground systems include multiple recovery mechanisms. Examples include hot swappable power supplies, disk arrays, network components, load balancers, virtual machine restarts, messaging middleware, and application specific mechanisms DT&E and OT&E cannot always be relied upon to provide sufficient data to measure the switchover probabilities for all such mechanisms. Where redundancy is used, switchover success probabilities must also be measured to assess the reliability of the redundant subsystem. To the extent possible, the RAM should conduct failure/recovery testing either as part of the DT&E and OT&E program or as a related activity before or after such testing.

31.4.10 Design and Milestone Reviews

Design and milestone reviews enable both internal and external stakeholders to assess the progress of a ground system and to identify issues based on artifacts and other data produced during the development process. The reviews and artifacts are defined by the program and the contract, but typically include six or seven major events. Table 31-3 shows a typical list of such events for ground systems, the major RAM-related events, the artifacts, and the issues to be addressed.

Table 31-3. RAM Artifacts and Issues by Milestone Reviews

Milestone or Design Review	Artifacts	Issues
System Specification Review (SSR)	System requirements Preliminary RAM allocations	• Completeness of RAM system-level requirements • Feasibility of top-level RAM allocations
System Design Review	Subsystem requirements Detailed RAM Allocations System architectures	• Completeness and traceability of subsystem RAM requirements • Feasibility of detailed RAM allocations (including software)
Preliminary Design Review	Preliminary RAM model System and Software architecture Preliminary software designs Preliminary failure modes and effects analysis	• Feasibility of design to meet requirements • Robustness of the software architecture with respect to failure detection, containment, and recovery • Identification of failure detection and recovery requirements for hardware and software components
Critical Design Review	Complete RAM model Completed hardware and software design Completed RAM modeling and prediction Completed failure modes and effects analysis Preliminary requirements verification and test plans	• Feasibility of design to meet requirements • Robustness of the software architecture with respect to failure detection, containment, and recovery • Identification of failure detection and recovery requirements for hardware and software components • Test program includes plans for both stability and failure recovery testing
Test Readiness Review	Discrepancy reports FRACAS data Test plans Test procedures	• Review of DRs for unanticipated failure modes • Review of FRACAS for resolution of failure trends • Test procedures provide sufficient data to assess failure probability' • Stability testing allows sufficient time for testing MTBF • Maintenance testing allows testing of all significant diagnostics
Informal and Formal and Tests	Test results DRs FRACAS	• Review of DRs for unanticipated failure modes • Review of FRACAS for resolution of failure trends

Milestone or Design Review	Artifacts	Issues
		• Assess test results for conformance with qualitative and quantitative RAM requirements
Independent Development and Operational Testing	Test objectives Test plans Test procedures Test results	

31.5 Key Lessons Learned

The following items are taken primarily from the software reliability appendix of the DOD RAM Guide [1] and the National Research Council [9]:

1. Ignoring RAM requirements can have a severe impact on operational effectiveness, suitability, and lifecycle cost. In a 2003 report, the U.S. Congress Government Accountability Office looked at five large systems built during this period. None of the programs prioritized reliability, maintainability, and availability. As a consequence, they found that once fielded, some systems were not achieving the readiness rates that program officials thought were possible during development [23]. A 2008 Defense Science Board Report noted that four out of seven large programs (known as Acquisition Category I, or ACAT 1) were deemed operationally unsuitable after independent operational test and evaluation (IOT&E) [24]. The low readiness rates and high sustainability costs impacted the ability of the DOD to acquire weapons systems.

2. Government requirements for system and software RAM contained in CDDs and TRDs should be operationally meaningful and attainable. These requirements should be developed during concept development based on mission needs and with lifecycle cost, affordability, and sustainment constraints taken into account [25].

3. RFP and contracting procedures should give appropriate attention to reliability in the RAM requirements and the resulting program. The RFP should include design-for-reliability activities that elevate the level of initial system reliability and RAM-focused test and evaluation events that provide comprehensive examinations of operational reliability and availability of the hardware and software [26].

4. Automated failure detection and recovery (also called fault tolerance) can't be "bolted on" in software systems. Such provisions must be integrated into the architecture definition and carried through into the

detailed design. The redundancy and switchover provisions in commercial software infrastructure products will not provide application-level fault tolerance. Reliability teams should be empowered to direct development organization design, development, and test activities to include failure detection and recovery provisions in their designs [26].

5. Software discrepancy reports should include symptom classifications. Such classifications to enable identification of trends and comparison against the expected failure behavior. When the failure behavior observed through the symptom classifications in the DRs deviates significantly from the failure detection and recovery provisions in the design, revisions should be made.

6. Metrics related only to software DRs do not provide sufficient insight into system reliability and availability because they do not account for operating time or restoration time. RAM metrics are related to times between failures and must include operating time. Where automated recovery provisions exist, recovery times and recovery probabilities should also be measured.

7. Test planning should include appropriate applications of reliability growth methodologies (i.e., compatible with underlying assumptions) for determining the time required for of system-level reliability testing and the validity of assessment results [26].

8. Sustained funding is needed throughout system definition, design, and development, to (a) incentivize development organization reliability initiatives and (b) accommodate planned reliability design and testing activities, including any revisions that may arise [26].

9. A troubled RAM program is frequently a leading indicator of a troubled program.

31.6 Government and Contractor Enabling Processes and Products

The primary government (acquisition organization) products to enable an effective RAM program for software intensive ground systems are the RFP and the resultant contract (including fee and incentive structures). The primary contractor (development organization) enabling processes emanate from the statement of work and the contractually defined deliverables are the enabling products. RFPs should ensure that software RAM considerations are addressed in both the offerors' proposals and in the subsequently written RAM and SEMPs during the initial stages of program execution. Such considerations include:

- System reliability models that can account for failure rates, automated failure detection and recovery times, recovery probabilities, and restoration times (if recovery fails) of both hardware and software components.

- Software architecture features that enable failure containment and recovery in defined regions (or components) in accordance with standard and robust (i.e., simple, verifiable, and effective) failure detection and recovery mechanisms.

- Characterization of the failure rates, recovery probabilities and recovery times of previously developed components by testing in an execution environment representative of what is expected during operation. Reused components include "hypervisors" (i.e., virtual machines that exploit the multi-core capabilities of server microprocessors), operating systems, protocol stacks, database management systems, language-specific virtual machines (e.g., Java virtual machines), application servers (if applicable), enterprise service buses (if applicable), object brokers (if applicable), messaging middleware, network management systems, and reused application software.

- Measurement of both the operating time and the number of failures of project-developed software to enable tracking of achieved reliability and reliability growth which is currently required by DoDI 5000.02.

- Characterization through testing of recovery times and recovery probabilities of architecture and application-specific failure detection and recovery features.

The RAM CDRL item contractual requirements are generally determined by the RAM standard incorporated into the contract. For ground systems, GEIA STD-009 [27] has been adopted by the DOD although some programs still used MIL-STD -785B [28]. All of these standards have a common set of core deliverables, which are shown in Table 31-4. The right column shows a summary of the tailoring to the RAM deliverables to include software

Table 31-4. Tailoring of RAM CDRLs to Address Software

CDRL Item	Tailoring for Software
RAM program plan	Explicitly address software as part of the reliability program in modeling and analysis, FRACAS, FMEA/CA, and test plans.
System reliability modeling and allocation reports	Include software in reliability, maintainability, and availability allocations. Include allocations for software recovery times and recovery probabilities for availability modeling (to increase availability, shortening recovery times can often be more easily achieved than increasing MTTFs).
Failure reporting, analysis, and corrective action system (FRACAS) plans and reports	Include software as well as hardware failures. Define the start of the software failure collection effort from the time that software leaves the control of the individual programmer (usually at software integration time).
failure modes and effects analysis/criticality analysis (FMEA/CA), critical items list (CIL) reports	Include software failure modes at the functional level FMEAs. Assess severity of effects for both hardware and software failures. Use allocated values of failure rates for criticality analyses; require inclusion of software as well as hardware components for the critical items list.
Reliability test plans and reports	Include software in the test plan; ensure that all software failure events are recorded (whether or not they are reproducible), operating times for all major software components are included, and that recovery times are recorded.

Just as the RAM program and CDRLs need to be tailored to address software, the software CDRLs need to be tailored to address system RAM issues. However, whereas RAM CDRLs relate primarily to analysis, reporting, and oversight, the software CDRLs should address the actual product being developed. Thus, the development phase must also be considered. Table 31-5 shows the tailoring of software deliverables by development phase to include RAM.

Table 31-5. Tailoring of Software CDRLs to Address RAM

Phase	CDRL Item	Tailoring for RAM
Requirements	Subsystem or element requirements specification	• Requirements clearly identify and fully define operational modes (normal mode, graceful degradation, safe mode, or other contingency modes)
	Software requirements specification	• Failure rates, recovery times, recovery probabilities (i.e., probability of effectiveness of automated recovery) , restoration times (i.e., time to restore failure given that an automated recovery is ineffective) allocated to the software components that will exist at runtime • Automated failure detection and recovery requirements defined including explicit criteria for what failure classes will be handled (for example, are only single-event failures handled?) • Timing requirements defined for both steady state and recovery—in particular, isolation, recovery, and repair durations (average and maximum) have been identified and allocated • Operator monitoring and control requirements to enable identification of software failures and restoration actions within the constraints of the MTTR requirements • Requirements for failure rates, failure detection capabilities, and recovery probability are derived as lower-level design elements are defined (not known at the beginning of the design).
	Interface requirements specifications	• Requirements for constraints on data and parameters in the interface • Requirements for indications of failures across the interface • Timing limits (violation of which would be a failure)
Software Architecture	System and software architecture description documents	• Definition of identifiable architecture components to manage failures in the system (replication, restart, message delivery assurance, synchronization of state data, integrity of database system, system status monitoring and reporting to sustainment personnel)
	Software architecture description	• Fault containment boundaries are explicitly and unambiguously defined, and the failure containment mechanisms at these boundaries are dependable
	Software architecture or software interface design document	• Protocols for processor health messages, heartbeats, and network traffic account for steady state and failure conditions
Detailed design and coding	Software design document or software development plan	• Design standards exist to enforce architectural provisions for failure containment, failure detection, and recovery (including interfaces, data types, exception handling, input and output validation)

Phase	CDRL Item	Tailoring for RAM
	Software development plan	• Software development procedures emphasize testing and recording of results of testing for both normal and off-normal cases (including automated regression testing)
Verification and Qualification	Software verification plan, software test plan, software test description	Verify that • All credible failure modes in lower-level SW, HW, and networks been identified • Traceability is established from the design to the system availability/maintainability requirements • The COTS/reuse products meet the reliability, availability, and fault management requirements (e.g., past service history, integration testing, etc.) • Detection and recovery provisions are defined for all identified failure modes addressed in the architecture • Failure detection/recovery durations meet the operational availability requirements
	Software architecture document and system reliability modeling and allocation reports	• Verify that software architecture, when combined with the hardware architecture, will meet system reliability and availability requirements (including maximum outage duration)
	Software test plans and descriptions	• Test program includes test cases verifying system failure detection and recovery capabilities using input from the FMEA/CA and FRACAS • Feedback from the testing program into the reliability analysis and model for measurement-based assessment of reliability/availability related parameters. • Observed failure modes are reconciled with anticipated failure
Operations	Various	• Software maintenance documentation, user manuals, operating, maintenance and test procedures, configuration data, interface documentation, and data rights are complete and sufficiently clear for use during sustainment • Monitoring and diagnostic capabilities allow identification and correction of software problems within MTTR constraints

31.7 Practice Task Application Example

This section that illustrates how software concerns are integrated into the contractual provisions associated with RAM [29]. The first subsection shows the tailoring of a Statement of Work for a ground system, the second shows tailoring of a standard to ensure that software reliability considerations are considered, and the third shows the tailoring of a Data Item Description to ensure that software issues are considered in an RAM-related contractual deliverable.

31.7.1 Sample Statement of Work

The following statement of work concerns the elements of the reliability program plan described above. References in parentheses are to CDRL items and applicable standards (in this case, MIL-STD-785B [28], MIL STD 1521B [30], MIL STD 1629A [14], and TR-RS-2013-00001 [33]). Of note is the fact that it is not necessary to insert software-specific modifications within the Statement of Work; they are necessary at lower levels, including standards tailoring (if a non-software specific standard such as MIL-STD-785B or 1629A is used) and for tailoring of Data Item Descriptions (DIDs) within the DD 1423 forms that define the requirements for Contract Data Requirements List (CDRL) items. Examples of standards and DID tailoring are shown in the next two subsections.

The Contractor shall:

(a) Develop a Failure Modes, Effects and Criticality Analysis, plan and execute a Failure Reporting and Corrective Action System (FRACAS) activity, and perform Reliability, Maintainability, and Availability tasks in accordance with MIL-STD 785B/T, MIL-STD 1629, and other applicable standards. (CDRL A021 [MIL-STD-785B (including Notice 1 and 2), TR-RS-2013-00001 [33], SEMP])

(b) Plan and perform reliability and maintainability analyses and studies linked to SSR, PDR, CDRs, and other appropriate events in the IMP. The Contractor shall allocate the contributions to system availability and dependability to OCX system elements and lower-level hardware and software items and components. ([MIL-STD-1521B (including Notice 1 and 2), MIL-STD-785B (including Notice 1 and 2), SSS, SSS (Prime Item, Critical Item Development Specification), SRS, Configuration Item Product Specification Prime Item/Critical Item, TR-RS-2013-00001 [33], OCX SEMP])

(c) Use quantitative methods to predict system availability and dependability. The Contractor shall use qualitative and quantitative methods to prevent design deficiencies and develop a robust system. (CDRL A069 [MIL-STD-785B (including Notice 1 and 2), TR-RS-2013-00001 [33], SEMP])

(d) Develop a RAM Program Plan. This plan shall include but not limited to: points of contact, resource plans, planned analysis, related CDRL deliverables, schedule, interaction and contribution of other related statement of work activities, reliability and availability growth projections at various key stages of DT&E, and key reporting milestones, a strategy for achieving reliability growth throughout DT&E

in order to meet RAM requirements. ([MIL-STD-785B (including Notice 1 and 2), TR-RS-2013-00001 [33], SEMP]).

(e) Address the RAM requirements when developing a signal monitoring architecture and MS design to meet specified operational availability and operational dependability requirements and legacy concepts in order to reduce system downtime and reduce overall sustainment costs for the MS as well as other OCX elements. ([DoDAF 1.5, MIL-PRF-49506 (including Notice 1), ANSI/AIAA R-100A, TR-RS-2013-00001 [33], MIL-STD-470B, SEMP]).

31.7.2 Tailoring of MIL-STD-785B to Include Software Reliability

Table 31-6, shows tailoring to MIL-STD-785B, which does not contain software-specific language (GEIA STD 009, its DOD-designated successor, does, as does TR-RS-2013-00001 [33], the reliability standard). The key conceptual modifications for software intensive ground systems are

- Substitution of "availability" for "reliability" because ground systems are continuously operated and the overall proportion of uptime is of primary importance

- Definition of the term "item" to include both hardware and software components.

- For the unit of analysis in reliability allocations, prediction models, and FMEAs, to specify that items at the lowest level of indenture include hardware Line Replaceable Units (LRUs) and software tasks or processes.

Table 31-6. Tailoring Example: MIL-STD-785B for Software Intensive Ground Systems

Task or Section or Paragraph or Sentence	Tailoring
Task 101 General	Substitute "reliability and availability" for "reliability"
Task 101.1.a,	For "SOW", substitute the following "Segment Specification – including hardware and software"
Task 101.1.b, final sentence	Add to the end of the sentence "for both hardware and software"
Task 101.1d, final sentence	Replace with the following: "The description shall specifically include the procedures to be employed which assure that applicable reliability data are derived from and traceable to both the software development and testing activities and the reliability tasks specified in the applicable LSAP and reported on the appropriate LSAR"

Task or Section or Paragraph or Sentence	Tailoring
Task 101.3	Replace with the following: “Contractor at their option may include the material specified in Task 104, par. 104.3.1 in the reliability/availability program plan.”
Task 102, paragraph 102.1, final sentence	Add “The plan for monitoring and controlling subcontractors and suppliers shall address all aspects of system reliability/availability, including hardware and software”
Task 103, paragraph 103.1, final sentence	Substitute “subsystem, equipment” with “segment, element, hardware, and software”
Task 104, paragraph 104.2,	Substitute “hardware or software” for the word “hardware”
Task 104, paragraph 104.3.1	Delete paragraphs a-e.
Task 104, paragraph 104.3.1	Replace with the following: “FRACAS to be defined in FRACAS plan by contractor with Government approval. FRACAS plan shall define FRACAS reporting and summary formats and how data will be collected and stored. FRACAS plan shall specify both hardware and software failures to be collected and describe the relationship to the software metrics report for software failures. FRACAS plan specify at what time software failure data will be collected and how reliability growth for both hardware and software will be accounted for. FRACAS plan specify how data on recovery times will be collected. FRACAS plan shall describe how failure data will be categorized to facilitate integration with allocations and models for the purposes of validation and FMEA analyses to identify unanticipated failure behaviors. FRACAS plan shall describe data content of incident records and supplementary information defining system configuration, operating phase, and effectiveness of mitigation measures (including recovery and repair)”
Task 201, general	Substitute “reliability and availability” for reliability
Task 201, paragraph 201.2.1	Substitute “system/subsystem/equipment” with “segment/element/hardware/software”.
Task 201, paragraph 201.2.1	Substitute “system/subsystem” with “segment/element.”
Task 201, paragraph 201.2.1, final sentence	Add “The contractor shall plan and document the modeling.”
Task 201, par. 201.2.3	Replace with the following: “Modeling techniques shall provide reliability and availability predictions that integrate allocated or predicted hardware and software failure rates, recovery times, recovery probabilities, operator/ maintainer fault isolation time, and repair time, and shall account for the multiple states possible within this complex integrated hardware/software system. Results shall be presented at the system, subsystem, and lower level elements as required.”
Task 202, paragraph 202.2.1	Add to the end of the paragraph “Allocation shall be made to software, hardware, and firmware.”
Task 202, paragraph 202.2.1, final sentence	Add “The contractor shall plan and document the allocation.”
Task 202, paragraph 202.3.1.a	Delete

Task or Section or Paragraph or Sentence	Tailoring
Task 202, paragraph 202.3.1, paragraph b	Replace with the following: "Allocations to be done to the software and hardware element level."
Task 203.2	Replace with the following: "Predictions shall be made at the level of LRUs and software executable that exist as processes at runtime. Predictions shall include failure rates, recovery times, recovery probabilities, and diagnostic effectiveness. Where commercial items or non-developmental items (hardware or software) are contemplated for the system design, these parameters rates shall be substantiated with measured or observed values. Substantiation can be done through (i) measurements, or (ii) analytical estimation/prediction or (iii) a combination of both. Where certainty on values of key parameters is not known, sensitivity studies shall be performed. For developmental items (hardware or software) without operational experience, either allocations or other methods approved by the PA shall be used. For such items, methods shall be described for verifying predictions or allocations using the data collected in the FRACAS or other testing or operational environments."
Task 204, paragraph 204.1	Add "both hardware and software failures and effects shall be analyzed. The capability of the system monitoring and diagnostics capabilities to identify each failure shall also be evaluated."
Task 204, paragraph 204.3.1, paragraph a	Replace with "The FMEA/CA shall be performed to the level of LRUs and executable software components that exist as processes at runtime."
Task 204, paragraph 204.3.1, paragraph b	Add "FMEA/CA format to be defined by the contractor and approved by the Government."
Task 204, paragraph 204.3.1, paragraph g	Add "FMEA/CA is to include software units or another software level of decomposition approved by the Government."
Task 204, paragraph 204.2.1, final sentence	Add "The contractor shall plan and document the results of the FMEA/CA."
Task 205	Delete
Task 206, paragraph 206.3.1, paragraph b	Remove and replace with "Items to be included in tolerance analysis to be defined by the contractor and approved by the Government."
Task 206, paragraph 206.2.1, final sentence	Add "The contractor shall plan and document the tolerance analysis in accordance with the Segment/Element Test and Analysis Report (CDRL A064, CDRL item e)."
Task 207	Delete
Task 209	Delete
Task Section 300 (Tasks 301 to 305)	Delete

31.7.3 Tailoring of a Data Item Description

Standard Data Item Descriptions (DIDs) that specify the contents of a RAM-related deliverable documents are frequently incomplete or ambiguous with respect to the inclusion of software. The example in this subsection, tailors language related to the Critical Items List of DI-ILSS-81495 (Failure Modes Effects and Criticality Analysis (FMECA) Report (30 OCT 1995). The Critical Items List is a list of components whose failure could critical impact the mission. The intent of identifying such items is to ensure that they have been subjected to a higher level of scrutiny during design, test, manufacturing (for hardware), shipping, and installation. The most significant changes to the DID are to:

- Define an "item" to include both hardware and software
- Use as a criterion for inclusion on the Critical Items List specific aspects of the mission which include controlling the constellation of satellite vehicles and providing a high integrity navigation signal.

The following is the text of the tailoring:

(a) The Critical Items List shall contain a listing of items (hardware or software) which can critically impact the constellation, any Satellite Vehicle (SV), navigation signal integrity or continuity, or the reliability or availability of any ground element. The list shall include summaries and references to specific documents defining compensating controls and features. The list is used to evaluate the adequacy and implementation of critical items controls. Critical Items are hardware, software, interface, or other items (referred to as "items") which require special attention because of complexity, application of state-of-the-art techniques, the impact of potential failure, or anticipated reliability problems. The following are typical circumstances which would cause an item to be included on a critical items list:

(1) A failure item that would seriously affect system operation or cause the system to not achieve mission objectives or meet system performance requirements for accuracy, availability, integrity, or continuity would represent a single point failure (i.e., any single hardware failure or software error which results in irreversible degradation of item mission performance below contractually specified levels).

(2) A failure of the item would prevent obtaining data necessary to evaluate accomplish of mission objectives.

(3) An item has exhibited an unsatisfactory operating history relative to required performance or reliability.

(4) An item does not have sufficient history or similarity to other items having demonstrated high reliability to provide confidence in its reliability.

(5) State-of-the-art techniques required to manufacture the item.

(6) Items are stressed in excess of criteria.

(7) The item has an operating, shelf-life, or environmental exposure limitation which warrants controlled storage or use.

(8) The item is known to require special processing, handling, transportation, storage or test.

(9) The item has exhibited an unsatisfactory operating history relative to required performance or reliability.

(10) The Item's past history, nature, function, or processing warrants total traceability.

(b) Critical Item list content. The list shall contain the following data:

(1) The identification of each critical item as cross-references to the related failure modes, effects, criticality analyses (FMECA), architecture and design data, hardware drawings (if appropriate), interface specifications, software documentation, aviation safety related analyses, system safety related analyses, schematics, and hardware.

(2) Reason or criteria causing the item to be classified as critical.

(3) A summary in specific terms for each critical item of the compensating features, controls, other practices incorporated or planned to minimize the likelihood or effect of the critical items failing during the life of the program (including but

not limited to specific documentation containing compensating features shall be referenced).

(4) Identification of the activity that discovered the critical items (FMECA test planning, stress analysis, reliability prediction or risk assessment), and reference to the related applicable documents.

(5) Rationale for not eliminating the critical item or related failure mode(s).

(6) Single point failure mode (SPFM) data.

(c) Supporting Data. The following information shall be included if it has not been previously submitted (such as in a System Safety Program Plan, planning and systems integration contract administration plan [PSICAP], fault hazard analysis [FHA], or investment capability analysis and assessment [ICAA]). If the supporting data is not included, a cross reference to where it appears shall be included in this section.

(1) A list of the criteria used to identify critical items (MIL-STD-1543B, Task 208, MIL-STD-1543B [36], and MIL-STD 882C [35] to provide guidance).

(2) A summary of the contractor's formal policy and procedures for critical item control and notification to affected personnel of the essential and critical nature of such items.

(3) A description of the traceability system applicable to the critical items list to facilitate follow-up verification that all planned critical item compensating features, controls, and practices have been implemented.

(4) A description of the methods and plans for updating the critical items list to provide timely management visibility. Identification of critical items which are on calendar age limited life and operating life item lists when applicable (Refers to MIL-STD-1543B, Task 208, paragraph 208.2.3)."

31.8 References

1. OUSD (AT&L(DS/SE/ED, DOD Guide For Achieving Reliability, Availability, and Maintainability, June 20, 2005, Page B-1, available at https://acc.dau.mil/CommunityBrowser.aspx?id=31008&lang=en-US.

2. Gray, J. "Why do computers stop and what can be done about it?" Tandem Computers, Tech. Rep. 85.7, 1985.

3. Lee, I. and R. Iyer. "Software dependability in the Tandem GUARDIAN system," IEEE Trans. Softw. Eng., vol. 21, no. 5, pp. 455–467, 1995.

4. Grottke, J.M., A. Nikora, and K. Trivedi, "An empirical investigation of fault types in space mission system software," in Proc. Intl. Conf. Dependable Systems and Networks., 2010, pp. 447–456.

5. Carrozza, Gabriella, Domenico Cotroneoy, Roberto Natellay, Roberto Pietrantuonoy, and Stefano Russoy. "Analysis and Prediction of Mandelbugs in an Industrial Software System" Software Testing, Verification and Validation (ICST), 2013 IEEE Sixth International Conference on (March 2013), pp. 262-271.

6. ISO/IEC/IEEE 24765, "Systems and software engineering—Vocabulary", 2010.

7. "IEEE Standards Dictionary: Glossary of Terms and Definitions," IEEE Std 100-2000.

8. DOD Guide For Achieving Reliability, Availability, and Maintainability, June 20, 2005, see footnote 12, Appendix C.

9. National Research Council Panel on Reliability Growth Methods for Defense Systems, "Reliability Growth: Enhancing Defense System Reliability", National Academies Press Washington, DC, 2015, www.nap.edu, section 10.

10. DoDI 5000.02, "Operation of the Defense Acquisition System, "January, 2015, Enclosure 3, Section 12, available at www.acq.osd.mil/fo/docs/500002p.pdf.

11. Chairman of the Joint Chiefs of Staff Instruction on the Joint Capabilities Integration and Development System, CJCSI 3170.01H, 10 January 2012, available from http://www.acqnotes.com/Attachments/CJCSI%203170.01H%20Joint%20

Capabilities%20Integration%20and%20Development%20System%2010%20January%202012.pdf.

12. Department of Defense, Reliability, Availability, Maintainability, and Cost Rationale Report Manual, June 1, 2009, available online at http://www.acq.osd.mil/se/docs/DoD-RAM-C-Manual.pdf.

13. Manual For The Operation Of The Joint Capabilities Integration and Development System, available at http://www.acqnotes.com/Attachments/JCIDS%20Manual%20for%20the%20Operation%20of%20the%20JCIDS%20%2019%20Jan%202012.pdf.

14. Lee, Inhwan, Dong Tang, Ravishankar K. Iyer, and Mei-Chen Hsueh. "Measurement-Based Evaluation of Operating System Fault Tolerance," IEEE Trans. Reliability, June 1993, Vol. 42, No. 2., pp. 238-249.

15. Tang, D. and H. Hecht. "An approach to measuring and assessing dependability for critical software systems," Proc. International Symp.On Software Reliability Engineering, 2-5 Nov. 1997, pp. 192 – 202.

16. MIL-STD-1629A, Procedures for Performing a Failure Mode, Effects, and Critical Analysis, November, 1980.

17. Public Law 111-123 "Weapons System Acquisition Reform Act of 2009," Section 102 (codified as 10 USC 4 Section 139d).

18. MIL HDBK 189 C – Reliability Growth Management , Army Materiel System Analysis Activity, June, 2011, available at www.dote.osd.mil/docs/dote-temp-guidebook/MIL-HDBK-189C.pdf.

19. Farr, William. "Software Reliability Modeling Survey", in *Handbook of Software Reliability Engineering* Edited by Michael R. Lyu, IEEE Computer Society Press and McGraw-Hill Book Company, 1996. Available at www.cse.cuhk.edu.hk/~lyu/book/reliability/.

20. Hecht, M. "Reliability Testing of Software Intensive Ground Systems" 28th Aerospace Testing Seminar, March 2014.

21. SMERFS (Statistical Modeling and Estimation of Software Reliability Functions), available at http://www.slingcode.com/smerfs/, 2002.

22. Computer Aided Software Reliability Estimation, 2000, available at https://www.openchannelsoftware.com/projects/CASRE_3.0.

23. DoDI 5000.02 Enclosure 3, section 12, paragraph c Enclosure 4, Section 4, paragraph (b)(7) and paragraph (f).

24. NIST/SEMATECH e-Handbook of Statistical Methods, section 8.3.11, available at http://itl.nist.gov/div898/handbook/apr/section3/apr311.htm.

25. United States General Accounting Office, "Best Practices: Setting Requirements Differently Could Reduce Weapon Systems' Total Ownership Costs", Report No.GAO-03-057, February 2003 available at http://www.gao.gov/assets/160/157396.pdf.

26. Seglie, Ernest. (Science Advisor, DOT&E), "Investing in Reliability, Availability, and Maintainability and the Effect on Logistics, Operational Support, and Lifecycle Cost", http://www.gardenstatesole.org/events/symposium08/seglie.pdf.

27. DOD Guide for Achieving Reliability, Availability, and Maintainability, June 20, 2005, see footnote 12, page B-2.

28. National Research Council Panel on Reliability Growth Methods for Defense Systems, "Reliability Growth: Enhancing Defense System Reliability", National Academies Press Washington, DC, 2015, www.nap.edu.

29. ANSI/GEIA-STD-0009, Reliability Program Standard for Systems Design Development and Manufacturing, November, 2008.

30. MIL STD 785B Military Standard: Reliability Program for Systems and Equipment Development and Production, September, 1980.

31. Next Generation Operational Control System (OCX) Fact Sheet, June, 2014, available at http://www.losangeles.af.mil/library/factsheets/factsheet.asp?id=18676.

32. MIL STD 1521B, Military Standard: Technical Reviews and Audits for Systems, Equipments, and Computer Software (04 JUN 1985).

33. Shaw, Brian. *Systems Engineering Requirements and Products*. TR-RS-2013-00001, The Aerospace Corporation, El Segundo, CA. February 28, 2013

34. Ingram-Cotton, John B., Myron Hecht, Roland J. Duphily. *Reliability Program for Space Systems*. TR-RS-2007-00013, The Aerospace Corporation, El Segundo, CA. July 01, 2007.

35. MIL-STD-882C. *System Safety Program Requirements*. 1993.

36. MIL-STD-1543B. *Reliability Program Requirements for Space and Launch Vehicles*. 1988.

31.9 Acronyms

ACAT	acquisition category
AIAA	American Institute of Aeronautics and Astronautics
ANSI	American National Standards Institute
AO	operational availability
AT&L	Acquisition, technology, and logistics
CA	criticality analysis
CASRE	computer-aided software reliability estimation
CDD	capability development document
CDR	critical design review
CDRL	contract data requirements list
CI	configuration item
CJCSI	chairman joint chiefs
COTS	commercial off-the-shelf
DBMS	data base management system
DID	data item description
DoDAF	Department of Defense, Air Force
DoDI	Department of Defense Instruction
DR	discrepancy reporting(s)
DT&E	design test and evaluation
FHA	fault hazard analysis
FMEA	failure modes and effects analysis
FMECA	failure modes and analysis
FRACAS	failure reporting and corrective action system
GEIA	Government Electronics and Information Technology Association
GOTS	government off-the-shelf
GPS	Global Positioning System
HW	hardware
ICAA	investment capability analysis and assessment
ILS	integrated logistics support
IMP	integrated master plan
IOT&E	independent operational test and evaluation
JCIDS	Joint Capabilities Integration and Development System
KPP	key performance parameters
KSA	key system attributes
LRU	line replaceable units
LSAP	logistic support analysis plan
LSAR	logistic support analysis record

MIL HDBK	military handbook
MTBF	mean time between failures
MTTF	mean time to failure
MTTR	mean time to restore
O&S	operations and sustainment
OCX	operational control system
OSS	open source software
OT&E	operational test and evaluation
PDR	preliminary design review
PSICAP	planning and systems integration contract administration plan
R&M	reliability and maintainability
RAM	reliability, availability, and maintainability
RAM-C	reliability, availability, maintainability, and cost
RFP	request for proposal
RMA	reliability, maintainability, and availability
SEMP	systems engineering management plan
SMERFS	statistical modeling and estimation of software reliability function
SOA	service oriented architecture
SSR	system specification review
SV	satellite vehicle
SW	software
TRD	technical requirements document
TRR	test readiness review
TT&C	telemetry, tracking, and control
USD	Under Secretary of Defense
WSARA	weapons systems acquisition reform act

Chapter 32
Cybersecurity

Marybeth S. Panock
Validation and Requirements
Cyber Acquisition and Validation Department

32.1 Introduction

Ground systems acquired as part of national security space (NSS) systems are largely information systems that are subject to cyber threats from both external and internal adversaries. These threats, if realized, can have a huge negative impact on the mission and function of a ground system by compromising the confidentiality, integrity, or availability of the information it handles. NSS ground systems historically have been unique to the mission with an independent set of applications, servers, and workstations, with associated local area networking connected via wide area networks to other related independent ground systems controlling other satellites and managing other missions. However, the paradigm is changing to a shared service architecture where the infrastructure, platform, and software services may be obtained from different service providers. The environments would likely be shared with other organizations and missions instead of being standalone systems in a cloud environment for host-compute-storage services. The threats, vulnerabilities, risks, and mitigations apply to both architectures and will have to be assessed with the configuration in mind.

The government has developed and adopted several policies to protect NSS systems, and several that apply to NSS Ground systems. The risk management framework (RMF), developed by the National Institute of Standards (NIST), and adopted and adjusted by the joint task force which included the Department of Defense (DOD), the Office of the Director of National Intelligence (ODNI), and the Committee on National Security Systems (CNSS), is a common federal information security framework developed to improve information security, strengthen risk management processes, and encourage reciprocity among federal agencies. A major advantage of the RMF process is its commonality across the federal government, e.g., DOD [1], the intelligence community, civilian agencies, and it is risk focused not compliance focused as the previous processes had become. The emphasis has changed from end-of-development assessment to early-on security engineering.

There are cybersecurity tasks required to implement RMF, given in the security controls catalog NIST SP 800-53 [2]. RMF is a six-step security lifecycle process for categorizing the sensitivity of the system, selecting the security controls to address the threat, implementing the security controls, assessing their

implementation, authorization, and monitoring the functioning of the security controls throughout operations.

DoDI 8510.01 March 2014 Risk Management Framework (RMF) for DOD Information Technology (IT) [3] reissues and renames DOD Instruction (DoDI) 8510.01 and implements

1. NIST SP 800-37, Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach, February 2010, as amended [1],
2. Subchapter III of chapter 35 of Title 44, United States Code (also known as the "Federal Information Security Management Act (FISMA) of 2002") [4],
3. Committee on National Security Systems (CNSS) Instruction 1253, Security Categorization and Control Selection for National Security Systems, March 15, 2012, as amended [5], and
4. NIST SP 800-53, Security and Privacy Controls for Federal Information Systems and Organizations, current edition [2].

by establishing the RMF for DOD IT (referred to in this instruction as "the RMF"), establishing associated cybersecurity policy, and assigning responsibilities for executing and maintaining the RMF.

The RMF replaces the DOD Information Assurance Certification and Accreditation Process (DIACAP) and manages the life-cycle cybersecurity risk to DOD IT in accordance with

1. NIST SP 800-53A, Guide for Assessing the Security Controls in Federal Information Systems and Organizations: Building Effective Security Assessment Plans, June 2010, as amended [6],
2. DOD Instruction 8500.01, Cybersecurity, March 14, 2014 [7], and
3. DOD Directive 8000.01, Management of the Department of Defense Information Enterprise, February 10, 2009 [8].

DoDI 8500.01 Cybersecurity [7]

1. reissues and renames DOD Directive (DoDD) 8500.01E [9] as a DOD Instruction (DoDI) to establish a DOD cybersecurity program to protect and defend DOD information and IT;
2. adopts the term "cybersecurity" as it is defined in National Security Presidential Directive-54/Homeland Security Presidential Directive-23 [10] to be used throughout DOD instead of the term "information assurance" (IA),
3. and is the base instruction which calls out DoDI 8510.01 [3].

CNSSI No. 1253 (2014) "Security Categorization and Control Selection for National Security Systems" [11]. Figure 32-1 shows that RMF is required by several government mandates, such as:

1. FISMA [4],
2. DoDI 5000.2 Operation of the Defense Acquisition System [12],
3. DoDI 5200.39 Critical Program Information (CPI) Protection Within the Department of Defense [13],
4. CNSSP No. 12, Committee on National Security Systems Policy (CNSSP) National Information Assurance Policy for Space Systems Used to Support National Security Missions [14], and
5. Intelligence Community Directive (ICD) 503 Information Technology Systems Security Risk Management, Certification and Accreditation [15].

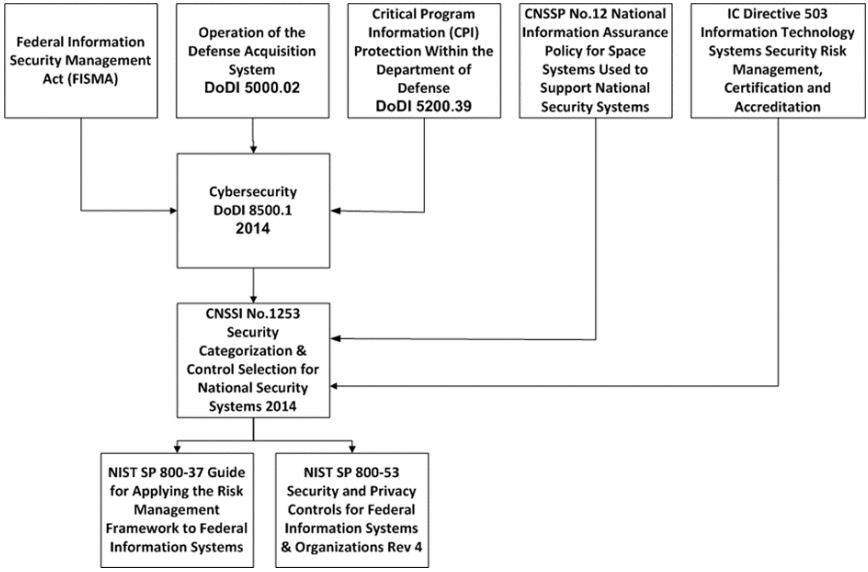


Figure 32-1. Cybersecurity governing document information flow.

32.2 Definitions

Adequate Security Security commensurate with the risk and the magnitude of harm resulting from the loss, misuse, or unauthorized access to, or modification of, information. This includes assuring that systems and applications used by the agency operate effectively and provide appropriate confidentiality, integrity, and availability through the use of cost-effective management, personnel, operational, and technical controls [16].

Advanced Persistent Threat An adversary that possesses sophisticated levels of expertise and significant resources which allow for opportunities to achieve objectives by using multiple attack vectors (e.g., cyber, physical, and deception). These objectives typically include establishing and extending footholds within the information technology (IT) infrastructure of the targeted organizations for purposes of exfiltrating information; undermining or impeding critical aspects of a mission, program, or organization; or positioning itself to carry out these objectives in the future. The advanced persistent threat: (a) pursues objectives repeatedly over an extended period of time, (b) adapts to defenders' efforts to resist it, and (c) is determined to maintain the level of interaction needed to execute objectives.

Assurance Measure of confidence that the security features, practices, procedures, and architecture of an information system accurately mediates and enforces the security policy [17].

Audit Log A chronological record of information system activities, including records of system accesses and operations performed in a given period [17].

Authentication Verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system [18].

Authorization: Access privileges granted to a user, program, or process or the act of granting those privileges [17].

Availability Ensuring timely and reliable access to and use of information. Source: [19]

Baseline Configuration A documented set of specifications for an information system, or a configuration item within a system, that has been formally reviewed and agreed on at a given point in time, and which can be changed only through change control procedures.

Boundary Protection Monitoring and control of communications at the external boundary of an information system to prevent and detect malicious and other unauthorized communications, through the use of boundary protection devices (e.g., gateways, routers, firewalls, guards, encrypted tunnels).

Confidentiality Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information [19].

Configuration Management A collection of activities focused on establishing and maintaining the integrity of information technology products and information systems, through control of processes for initializing, changing, and

monitoring the configurations of those products and systems throughout the system development lifecycle.

Countermeasures Actions, devices, procedures, techniques, or other measures that reduce the vulnerability of an information system. Synonymous with security controls and safeguards [17].

Cyber Attack An attack, via cyberspace, targeting an enterprise's use of cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of the data or stealing controlled information [17].

Cybersecurity The ability to protect or defend the use of cyberspace from cyber attacks [17].

Incident An occurrence that actually or potentially jeopardizes the confidentiality, integrity, or availability of an information system or the information the system processes, stores, or transmits or that constitutes a violation or imminent threat of violation of security policies, security procedures, or acceptable use policies [18].

Information Security The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability [19].

Information Security Policy Aggregate of directives, regulations, rules, and practices that prescribes how an organization manages, protects, and distributes information [17].

Information System A discrete set of information resources organized for the collection, processing, maintenance, use, sharing, dissemination, or disposal of information [20, 21].

Information System Resilience: The ability of an information system to continue to: (a) operate under adverse conditions or stress, even if in a degraded or debilitated state, while maintaining essential operational capabilities, and (b) recover to an effective operational posture in a time frame consistent with mission needs.

Information System-related Security Risks Information system-related security risks are those risks that arise through the loss of confidentiality, integrity, or availability of information or information systems and consider impacts to the organization (including assets, mission, functions, image, or reputation), individuals, other organizations, and the Nation.

Integrity Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity [19].

Malicious Code Software or firmware intended to perform an unauthorized process that will have adverse impact on the confidentiality, integrity, or availability of an information system. A virus, worm, Trojan horse, or other code-based entity that infects a host. Spyware and some forms of adware are also examples of malicious code.

Mobile Code Software programs or parts of programs obtained from remote information systems, transmitted across a network, and executed on a local information system without explicit installation or execution by the recipient.

Role-Based Access Control Access control based on user roles (i.e., a collection of access authorizations a user receives based on an explicit or implicit assumption of a given role). Role permissions may be inherited through a role hierarchy and typically reflect the permissions needed to perform defined functions within an organization. A given role may apply to a single individual or to several individuals.

Security A condition that results from the establishment and maintenance of protective measures that enable an enterprise to perform its mission or critical functions despite risks posed by threats to its use of information systems. Protective measures may involve a combination of deterrence, avoidance, prevention, detection, recovery, and correction that should form part of the enterprise's risk management approach [17].

Security Policy A set of criteria for the provision of security services [17].

Security Requirements: Requirements levied on an information system that are derived from applicable laws, Executive Orders, directives, policies, standards, instructions, regulations, procedures, or organizational mission/business case needs to ensure the confidentiality, integrity, and availability of the information being processed, stored, or transmitted [18].

Threat: Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, or the Nation through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service [17].

Vulnerability: Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source. Source: [17]

32.3 Risk Management Framework and Critical Tasks

Cybersecurity should be integrated in the ground system engineering development process from the beginning; primarily to ensure that the system enjoys the confidentiality, integrity, and availability critical for mission success. The mission will not succeed if the data can't be trusted, if the system is not available, or if the adversary can view the critical information. The RMF fosters and enables the integration of cybersecurity into the system development lifecycle (SDLC). The RMF provides a disciplined and structured process that integrates cybersecurity and risk management activities into the major phases of the SDLC; from initiation (concept/requirements definition), through development, implementation, and finally operations.

The RMF steps are:

- Categorize the information system and the information processed, stored, and transmitted by that system based on an impact analysis of the loss of confidentiality, integrity, and availability of the information.
- Select an initial set of security controls for the information system based on the security categorization; tailor and supplement the security control set as needed based on an organizational assessment of risk and local conditions.
- Implement the security controls and describe how the controls are employed within the information system and its environment of operation.
- Assess the security controls using appropriate assessment procedures to determine the extent the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting the security requirements for the system.
- Authorize information system operation based on a determination of the risk to organizational operations and assets, individuals, other organizations, and the nation resulting from the operation of the information system and the decision that this risk is acceptable.
- Monitor the security controls in the information system on an ongoing basis including assessing control effectiveness, documenting changes to the system or its environment of operation, conducting security impact analyses of the associated changes, and reporting the security state of the system to designated organizational officials.

As noted earlier, the architecture of the ground system may be a standalone or a shared service/cloud environment. This affects which organization is responsible for the security controls. In a traditional standalone ground system, the individual system or program of record is responsible for all the applicable 'selected' controls that are not inherited from the physical environment where it

is hosted. In a shared service environment, the service providers at the host, platform, or software application level may provide the security services that will satisfy some of the controls. Both agency-specific documents and CNSSI 1253 identify controls that are potentially common or inheritable that are developed, implemented, assessed, and monitored by a provider organization [11]. In addition, some agency specific documents include recommendations for which controls can be hybrid controls, i.e., controls implemented in part as a common control and in part as a system-specific control. However, there are system-specific security controls not designated as common security controls nor are they a portion of a hybrid control that is to be implemented within an information system. While the program of record is responsible for knowing how each of the applicable 'selected' controls are satisfied, they always responsible for implementing (and verifying, assessing, etc.) the system-specific security controls.

The RMF also specifies tasks to be accomplished "concurrently with or as part of system development lifecycle processes, taking into account appropriate dependencies. This helps to ensure that organizations are effectively integrating the process of managing information system-related security risks with system development lifecycle processes." The process of implementing the RMF tasks (i.e., the order and manner in which the tasks occur and are executed, the names of primary/supporting roles, the names and format of artifacts) may vary from organization to organization. The RMF tasks can be applied at appropriate phases in the system development lifecycle. While the tasks appear in sequential order, there can be many points in the risk management process that require divergence from the sequential order including the need for iterative cycles between tasks and revisiting tasks" [1].

The Aerospace Corporation has identified critical tasks at acquisition program milestones and major events, to ensure uniformity across programs, to foster consistency, to provide a tool for assessing program preparedness, and to identify program risk. The cybersecurity tasks are focused on ensuring the cybersecurity is well integrated into the acquisition process, specifically the system development lifecycle phases with tasks for requirements, architecture, design, development, and verification. The SDLC phases are initiation, development, implementation, and operations. The tasks for each phase and milestone checkpoints are detailed.

32.3.1 Initiation Phase

The initiation phase of the SDLC which includes concept and requirements (high-level) definition is when the first two steps of the RMF process generally take place. These two steps are categorize and select. The categorize step identifies the impact levels for confidentiality, integrity, and availability based on the system and data in its intended environment. The system ends up with a

categorization of High-High-High for confidentiality, integrity, and availability, or it ends up with a categorization of High-Moderate-Low, or any such combination in between. This categorization leads to the selection of the baseline set of cybersecurity controls that will be applicable to the system at that particular level.

32.3.1.1 Categorize Information System (RMF Step 1)

There is one cybersecurity task in the categorization phase: assess system categorization. This task is supported by the following RMF tasks and milestone checkpoint questions.

RMF Categorization Tasks

- Categorize the information system and document the results of the security categorization in the security plan.
- Describe the information system (including system boundary) and document the description in the security plan.
- Register the information system with appropriate organizational program/management offices.

Milestone Categorization Checkpoint Questions

- Has the organization completed a security categorization of the information system (informed by the initial risk assessment) including the information to be processed, stored, and transmitted by the system?
- Are the results of the security categorization process for the information system consistent with the organization's enterprise architecture and commitment to protecting organizational mission/business processes?
- Do the results of the security categorization process reflect the organization's risk management strategy?
- Has the organization adequately described the characteristics of the information system?
- Has the organization registered the information system for purposes of management, accountability, coordination, and oversight?

32.3.1.2 RMF Select Security Controls (RMF Step 2)

There is one cybersecurity task in the selection step: assess cybersecurity control selection. This task is supported by the following RMF tasks and milestone checkpoint questions.

RMF Selection Tasks

- Identify the security controls that are provided by the organization as common controls for organizational information systems and document the controls in a security (or equivalent document).
- Select the security controls for the information system and document the controls in the security plan. The controls selected include the baseline controls for the identified impact levels as well as those from the applicable overlay, e.g., space, cross domain solution, intelligence, or classified data. Controls applicable to the system are then identified.
- Develop a strategy for the continuous monitoring of security control effectiveness and any proposed or actual changes to the information system and its environment of operation.
- Review and approve the security plan.

Milestone Selection Checkpoint Questions

- Has the organization allocated all security controls to the information system as system-specific, hybrid, or common controls?
- Has the organization used its risk assessment (either formal or informal) to inform and guide the security control selection process?
- Has the organization identified authorizing officials for the information system and all common controls inherited by the system?
- Has the organization tailored the baseline security controls to ensure that the controls, if implemented, adequately mitigate risks to organizational operations and assets, individuals, other organizations, and the nation?
- Has the organization addressed minimum assurance requirements for the security controls employed within and inherited by the information system?
- Has the organization consulted information system owners when identifying common controls to ensure that the security capability provided by the inherited controls is sufficient to deliver adequate protection?
- Has the organization supplemented the common controls with system-specific or hybrid controls when the security control baselines of the common controls are less than those of the information system inheriting the controls?
- Has the organization documented the common controls inherited from external providers?
- Has the organization developed a continuous monitoring strategy for the information system (including monitoring of security control effectiveness for system-specific, hybrid, and common controls) that

reflects the organizational risk management strategy and organizational commitment to protecting critical missions and business functions?
- Have appropriate organizational officials approved security plans containing system-specific, hybrid, and common controls?

32.3.2 Development Phase

The development phase includes parameter completion for the controls where the specifics are left to the organization, as well as two supporting RMF tasks, implement (and document) and assess. During this phase, the system security architecture and design are developed; application, operating system, and network security are determined; security unit testing takes place; and the security of the system is verified.

32.3.2.1 RMF Implementation (RMF Step 3)

There are nine cybersecurity tasks in the development-implementation phase which mirror the SDLC with tasks specific to encryption protection.

1. Assess Cybersecurity Requirements
2. Assess Cybersecurity Architecture
3. Assess Cybersecurity Design
4. Assess Cyber Application Security
5. Assess Cyber Operating System Security
6. Assess Cyber Network Security
7. Assess Cybersecurity Implementation and Unit Testing
8. Assess Cybersecurity Cryptographic Product Acquisition
9. Assess Cybersecurity Key Management Process

These tasks are supported by the following RMF tasks and milestone checkpoint questions.

RMF Implementation Tasks

- Implement the security controls specified in the security plan.
- Document the security control implementation, as appropriate, in the security plan, providing a functional description of the control implementation (including planned inputs, expected behavior, and expected outputs).

Milestone Implementation Checkpoint Questions

- Has the organization allocated security controls as system-specific, hybrid, or common controls consistent with the enterprise architecture and information security architecture?
- Has the organization demonstrated the use of sound information system and security engineering methodologies in integrating information technology products into the information system and in implementing the security controls contained in the security plan?
- Has the organization documented how common controls inherited by organizational information systems have been implemented?
- Has the organization documented how system-specific and hybrid security controls have been implemented within the information system taking into account specific technologies and platform dependencies?
- Has the organization taken into account the minimum assurance requirements when implementing security controls?

32.3.2.1.1 Requirements

The requirements phase translates the conceptual aspect of a system into a set of measureable, observable, and testable requirements. As with all functional requirements, the cybersecurity requirements must cascade through the acquisition software and system specification trees. The purpose of these requirements is to address the cyber threats, and to identify and comply with the relevant policies, at the DOD, agency, and national levels. With the commencement of the RMF process, all NSS systems must select their applicable cybersecurity controls from NIST SP 800-53 [2]. The selection process includes assignment of organization-defined parameters such as password length, allowed communications protocols, authorized source and destination addresses, etc. CNSSI 1253 interprets NIST SP 800-53 for national security systems by selecting controls for each of the security objectives impact levels and completing the parameters [1, 5]. In addition, other government organizations, such as in the intelligence community, have also selected applicable controls and specified the organization-specific parameters, in agency or organization specific directives. The selected controls must then be translated and catalogued as distinct requirements in the acquisition documents and CDRLs. The functional security requirements are selected from the controls imposed on the information system (as opposed to the organization) such as those in the access control (AC), audit and accountability (AU), configuration management (CM), contingency planning (CP), identification and authentication (IA), system and Communications Protection (SC), and System and Information Integrity (SI) families.

32.3.2.1.2 Architecture

The architectural stage of development focuses on the security architecture. Security architecture means the process of selecting design elements and principles to match a defined security need. In other words, the security architecture can be thought of as a body of high-level design principles such as least privilege that aim to limit the impact of even successful attempts to subvert software. The security architecture serves as the framework for secure design, which enables the four classic sets of cybersecurity controls, those that protect, deter, detect, and correct or mitigate the attacks on the confidentiality, integrity, and availability of the system. The role of the security architecture applies throughout the development process, and serves as the foundation of good software security engineering, which is critical for the development of secure software. The architecture phase identifies and defines the external interfaces, the cybersecurity requirements for those interfaces, establishes that the data received over those external interfaces will be properly validated, and that the cybersecurity requirements are properly allocated across the elements of the architecture. See NIST Special Publication 800-64 Revision 2, *Security Considerations in the System Development Lifecycle*, Section 2.1.4, Security Architectures [21].

32.3.2.1.3 Design

Good security design defends the system and applications from subversion or misuse, and protects the network and the information on it from internal and external threats. It provides a secure foundation for future extensions and software maintenance. Bad design makes the system more vulnerable to attack and more difficult to defend. The security design ensures that the external interfaces are adequately defended, that input passing through from these external interfaces is properly validated, that the security mechanisms are designed into specific software functions, and that the security policy has been decomposed into well-formed requirements. The security design addresses inputs, responses, exception conditions, error returns, and plans for validating input.

There should be a cybersecurity analysis activity with the objective of ensuring that the cybersecurity requirements are well understood and incorporated into the software and system design.

32.3.2.1.4 Application Security

Application security refers not only to the security and assurance of the security mechanisms but to the security and assurance of the whole software application, including developed code, legacy code, free and open source software (FOSS), and third party COTS products. (Application refers to the software that

implements the mission and is installed on the operating system.) Secure development practices foster application security. These include having a secure development standard, training developers to code to that standard, and conducting secure code reviews and/or peer reviews to that standard, either manually or with automated tools, e.g., SAST, DAST, or IAST. Aerospace TOR-2013-00742 *Secure Software Assurance Coding Guidance*, provides suggested topics and criteria for a secure coding standard [22]. The topic areas discussed include: trust boundaries, session management, input validation, output encoding, error and exception handling, failing securely, database security, server side controls, and race conditions. The NIST SP 800-53 Rev 4 System and Services Acquisition (SA) family of controls, most notably, SA-15 Development Process, Standards, and Tools, and SA-11 Developer Security Testing and Evaluation, address the topic of secure development and support 'application security [2].'

32.3.2.1.5 Operating System Security

The security of the operating system is the foundation for secure processing. The operating system needs to be securely configured in order to protect the application, the network interfaces, and itself. Exactly what the secure configuration of the operating system will be should not be decided in a vacuum; it should not be determined after development is complete and any unused ports and protocols, services and daemons removed. It should be decided upon before development starts by consensus among the cybersecurity staff, the system administrators responsible for OS operations and maintenance, and the software development team who will be developing the application on it. Each of these stakeholders should represent their requirements and constraints. Once agreed upon, it should be documented and verified. Software development should take place on this secure configuration, and the application should be verified on this secure configuration. The secure configuration should aim to enforce least privilege on processes and user classes, have virus protection, and an effective patch management process. NIST SP 800-53 control, CM-6 Configuration Settings, specifies that the configuration settings be established and documented using organization-defined security configuration checklists for information systems that reflect the most restrictive mode consistent with operational requirements [2]. The organization-defined security configuration checklist can be, for example, DISA Security Technical Implementation Guides (STIG) or NSA Systems and Network Attack Center (SNAC) hardening guidelines [23]. Topics that need to be addressed include system configuration, account configuration, file ownership and permissions, permitted programs and services, utilities, devices, commands, security patching, and anti-virus. In addition to configuration management, NIST SP 800-53 SA control, SA-4(7) Acquisition Process–NIAP Approved Protection Profiles, specifies that Information Assurance (IA)-enabled products be limited to those that have been successfully evaluated against a National Information Assurance partnership

(NIAP)-approved protection profile [2]. Operating systems are IA-enabled devices, and an evaluated one can be securely configured.

32.3.2.1.6 Network Security

Network security covers, at a top level, boundary defense, computer network defense, and the protection of external interfaces, as well as more detailed areas such as secure DNS, and the protection of communications. The NIST SP 800-53 control families that address this area include SC and system and information integrity (SI). Some example controls in this area include SC-07 Boundary Protection, SC-08 Transmission Confidentiality and Integrity, SC-10 Network Disconnect, SC-20 Secure Name/Address Resolution Service, SC-40 Wireless Link Protection, and SC-41 Port and I/O Device Access [2]. It is important, however, that the development team understands the requirements of secure networking and strive to use only secure protocols; e.g., SSH instead of telnet, Secure FTP instead of FTP or TFTP, and SSL or HTTPS instead of HTTP as defined in the secure configuration guides governing their system components. Network security is the first line of defense against cyber attacks and is therefore often a main focus of the assessment and authorization verification efforts.

32.3.2.1.7 Implementation and Unit Testing

Development, or implementation, is the process of creating and configuring the software that will satisfy cybersecurity requirements. As previously stated, the cybersecurity requirements need to have traceability within the acquisition software specifications and related contract and requirements list (CDRL) documents. There must be bidirectional traceability between the cybersecurity requirements and the cybersecurity code, and between the cybersecurity design units and the cybersecurity code units. Cybersecurity implementation processes need to be documented, and followed. There needs to be secure coding standards, along with practices, procedures, and conventions that are documented and followed.

Code, whether newly developed, reused, or open source (FOSS) must be consistent with the architecture and design, and correctly implemented following the cybersecurity implementation processes, and peer reviewed with all action items closed. Bugs in the implementation of the security mechanisms must be treated as any other software bug; tracked, fixed, reverified. The code must adhere to the program's coding standards, including the Secure Coding Standard, practices, procedures, and conventions.

Code peer reviews that assess the code against the contractor's coding standards, including the Secure Coding Standard, should be attended by qualified contractor personnel and, if allowed by contractor processes, qualified Aerospace personnel. The peer review team can include a cybersecurity subject

matter expert (SME) who would be responsible for reviewing the code to the Secure Coding Standard. Alternatively, the secure code review process can be distinct and in parallel, augmented by automated tools, such as Static Code Analysis Tools (SAST), Dynamic Code Analysis Tools (DAST), or Interactive Code Analysis Tools (IAST) [24]. These peer reviews need to be effective and formally handled, if not as contract deliverables to the government, but formal enough within the contractor organization that they benefit the developer and the code and result in better, more secure code that is resistant to cybersecurity threats. The peer review process should include developer/reviewer checklists based on the Secure Coding Standard, documented reports to the developer showing where the standard violations or security mechanism implementation error occurred, remediation recommendations, be tracked, remediated and peer reviewed again if severity of the violation or implementation error warrants.

32.3.2.1.8 Cryptographic Product Acquisition

Cryptographic product acquisition is not covered here in great detail as this is a very specialized area which requires a great deal of knowledge and coordination with NSA—expertise generally outside of that resident at Aerospace. They are covered in the MAB for awareness purposes but are not specifically part of the RMF process. For more information see the Cybersecurity MAB Level 2 tasks and references [25].

32.3.2.1.9 Key Management Process

The key management process is not covered here in great detail as this is a very specialized areas which requires a great deal of knowledge and coordination with the National Security Agency (NSA)—expertise generally outside of that resident at Aerospace. They are covered in the MAB for awareness purposes but are not specifically part of the RMF process. For more information see the Cybersecurity MAB Level 2 tasks and references [25].

32.3.3 Implementation Phase

During the implementation phase the system is built and installed in the operational environment. The RMF step is assessment. There are three supporting cybersecurity tasks in the implementation phase which define responsibilities for the assessment process: assess cybersecurity verification test planning, assess cybersecurity verification, and assess cybersecurity system assessment activities.

32.3.3.1 Assessment (RMF Step 4)

The most important thing to consider in cybersecurity requirements verification and test is that these requirements need to be verified in the acquisition process

along with the other functional requirements. It is not sufficient to rely on the assessment and authorization process (formerly the certification and accreditation process). The assessment and authorization (A&A) process occurs too late; the later you fix a problem the more costly it is. This has been documented in many quality assurance studies. Secondly, if the A&A process does identify secure software vulnerabilities they may not be cycled back to development phase to fix but may be sent to the operations phase to address. Operations may have other priorities or insufficient budget to address security vulnerabilities. Mitigations coded into the deployed operational software may not be rolled into the next software release and may reoccur. In the past when a compliance mind-set was pervasive and not a security risk mitigation perspective, there was the strong possibility that that A&A process would not adequately address the software application level, preferring to only assess the operating system and network level which can be handled with automated scanning tools. The emphasis was also primarily on the interconnection of systems and not on the internal system workings which has become increasingly more important.

During cybersecurity verification requirements and testing should be treated like any other engineering discipline. Test databases need to be adequate for the security qualification testing so that boundary conditions can be checked and negative testing can take place. It is not sufficient to verify that an operator can perform his intended function; it is necessary to verify that he can perform his job functions and the functions of no other roles, and mission operators in other roles cannot perform his.

The qualification environments need to adequate for the planned cyber testing, and the testing processes, methods, and tools must be appropriate to the cyber tasks. The testing products must be adequate for verifying both the functional and assurance requirements. There should be adequate discrepancy reports for the test findings. Program office personnel with cybersecurity expertise should participate in peer reviews, test readiness reviews, witness qualification testing, and participate in test exit reviews.

The cybersecurity verification should include independent assessment of the critical requirements, independent analysis of the discrepancy reports, and the planned mitigation. The status of the verification should be maintained and configuration managed (Note that CA-2 security assessments specify that there be a formal plan for assessment of the security controls of the system. Enhancement 1 to this control specifies independence.)

The assessment tasks are further supported by the following RMF tasks and milestone checkpoint questions.

RMF Assessment Tasks

- Develop, review, and approve a plan to assess the security controls.
- Assess the security controls in accordance with the assessment procedures defined in the security assessment plan
- Prepare the security assessment reporting documenting the issues, findings, and recommendations from the security control assessment.
- Conduct initial remediation actions on security controls based on the findings and recommendations of the security assessment report and reassess remediated control(s), as appropriate.

Milestone Assessment Checkpoint Questions

- Has the organization developed a comprehensive plan to assess the security controls employed within or inherited by the information system?
- Was the assessment plan reviewed and approved by appropriate organizational officials?
- Has the organization considered the appropriate level of assessor independence for the security control assessment?
- Has the organization provided all of the essential supporting assessment-related materials needed by the assessor(s) to conduct an effective security control assessment?
- Has the organization examined opportunities for reusing assessment results from previous assessments or from other sources?
- Did the assessor(s) complete the security control assessment in accordance with the stated assessment plan?
- Did the organization receive the completed security assessment report with appropriate findings and recommendations from the assessor(s)?
- Did the organization take the necessary remediation actions to address the most important weaknesses and deficiencies in the information system and its environment of operation based on the findings and recommendations in the security assessment report?
- Did the assessor reassess the remediated controls for effectiveness to provide the authorization official with an unbiased, factual security assessment report on the weaknesses or deficiencies in the system?
- Did the organization update appropriate security plans based on the findings and recommendations in the security assessment report and any subsequent changes to the information system and its environment of operation?

32.3.4 Operations Phase

At the completion of the implementation phase, the system is authorized based on the results of the assessment activities. Once the system is authorized (RMF Step 5) operational security monitoring begins (RMF Step 6).

32.3.4.1 Authorization (RMF Step 5)

For the RMF Authorization step, there is one cybersecurity task: Assess Cybersecurity System Authorization Activities. The supporting RMF tasks and the milestone checkpoint questions relate to the authorization of the information system.

RMF Authorization Tasks

- Prepare the plan of action and milestones based on the findings and recommendations of the security assessment report excluding any remediation actions taken.
- Assemble the security authorization package and submit the package to the authorizing official for adjudication.
- Determine the risk to organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, or the nation.
- Determine if the risk to organizational operations, organizational assets, individuals, other organizations, or the nation is acceptable.

Milestone Authorization Checkpoint Questions

- Did the organization develop a plan of action and milestones reflecting organizational priorities for addressing the remaining weaknesses and deficiencies in the information system and its environment of operation?
- Did the organization develop an appropriate authorization package with all key documents including the security plan, security assessment report, and plan of action and milestones (if applicable)?
- Did the final risk determination and risk acceptance by the authorizing official reflect the risk management strategy developed by the organization and conveyed by the risk executive (function)?
- Was the authorization decision conveyed to appropriate organizational personnel including information system owners and common control providers?

32.3.4.2 Monitoring (RMF Step 6)

During the operations phase the system is operational. There are five cybersecurity tasks during the operation phase to ensure that the system remains secure as it evolves and is used.

1. Assess Monitoring of Cybersecurity Controls in Operations
2. Assess Monitoring of Application Security in Operations
3. Assess Monitoring of Cyber Operating System Security in Operations
4. Assess Monitoring of Cyber Network Security in Operations
5. Assess Updates to Security Authorization Package is adequate

The supporting RMF tasks and milestone questions relate to the continuous monitoring of the security controls to assure that they are operating securely and effectively.

RMF Monitoring Tasks

- Determine the security impact of proposed or actual changes to the information system and its environment of operation.
- Assess the technical, management, and operational security controls employed within and inherited by the information system in accordance with the organization-defined monitoring strategy.
- Conduct remediation actions based on the results of ongoing monitoring activities, assessment of risk, and outstanding items in the plan of action and milestones.
- Update the security plan, security assessment report, and plan of action and milestones based on the results of the continuous monitoring process.
- Report the security status of the information system (including the effectiveness of security controls employed within and inherited by the system) to the authorizing official and other appropriate organizational officials on an ongoing basis in accordance with the monitoring strategy.
- Review the reported security status of the information system (including the effectiveness of security controls employed within and inherited by the system) on an ongoing basis in accordance with the monitoring strategy to determine whether the risk to organizational operations, organizational assets, individuals, other organizations, or the Nation remains acceptable.
- Implement an information system disposal strategy, when needed, which executes required actions when a system is removed from service.

Milestone Monitoring Checkpoint Questions

- Is the organization effectively monitoring changes to the information system and its environment of operation including the effectiveness of deployed security controls in accordance with the continuous monitoring strategy?
- Is the organization effectively analyzing the security impacts of identified changes to the information system and its environment of operation?
- Is the organization conducting ongoing assessments of security controls in accordance with the monitoring strategy?
- Is the organization taking the necessary remediation actions on an ongoing basis to address identified weaknesses and deficiencies in the information system and its environment of operation?
- Does the organization have an effective process in place to report the security status of the information system and its environment of operation to the authorizing officials and other designated senior leaders within the organization on an ongoing basis?
- Is the organization updating critical risk management documents based on ongoing monitoring activities?
- Are authorizing officials conducting ongoing security authorizations by employing effective continuous monitoring activities and communicating updated risk determination and acceptance decisions to information system owners and common control providers?

32.4 Cybersecurity Documentation

The RMF is designed to be complementary to and supportive of DOD's acquisition management system activities, milestones, and phases. RMF activities should be initiated as early as possible in the DOD acquisition process to increase security and decrease cost. Requirements development, procurement, and test and evaluation (T&E) processes should be considered in applying the RMF to the acquisition of DOD IT. Threats to these systems should be designated consistent with the most severe risk to any individual component or subcomponent for consideration of requirements, acquisition, and testing and evaluation. Figure 32-2 illustrates the alignment of RMF steps to the DOD acquisition lifecycle [3].

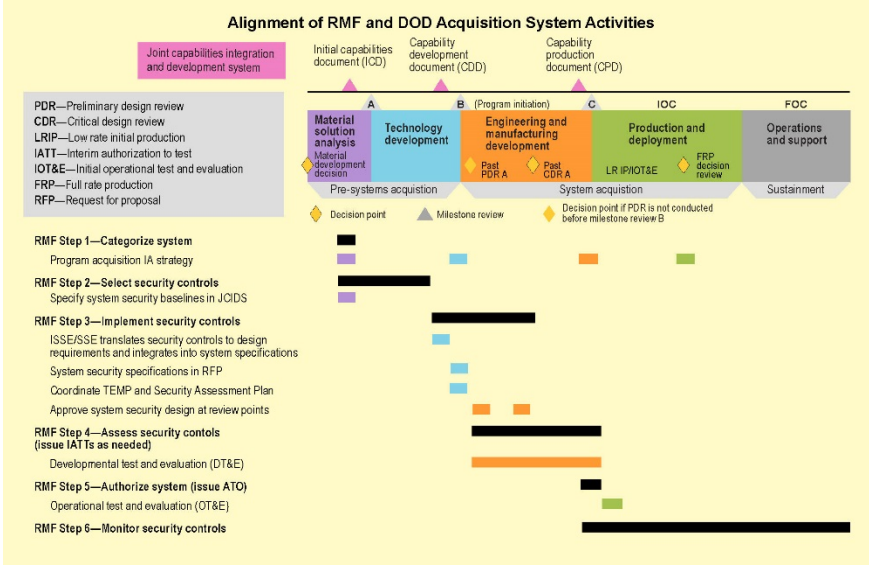


Figure 32-2. RMF and the defense acquisition management system.

The RMF process which encompasses the system development lifecycle specifies that a number of security documents be produced. The first and most important of these is the system security plan, which is described Table 32-1, along with several other documents that are generally required by the authorizing official (AO). The information in the security authorization package which contains the (a) the system security plan; (b) the security assessment report; and (c) the plan of action and milestones (POA&M). is used by authorizing officials to make risk-based authorization decisions.

Table 32-1. Security Documents Produced for RMF Assessment and Authorization

Title	Description
System Security Plan, NIST SP 800-18 [26]	Provides an overview of the security requirements for an information system and describes the security controls in place or planned for meeting those requirements
Security Assessment Report (SAR)	Provides visibility into specific weaknesses and deficiencies in the security controls employed within or inherited by the information system that could not reasonably be resolved during system development or that are discovered post-development

Title	Description
Risk Assessment Report (RAR)	Summarizes information on threats, vulnerabilities, and potential impacts, along with recommendations for risk mitigation. It provides the organizationally established level of acceptable risk associated with the operation of an IT system at a particular security level and identifies risk and assessed residual risk level for the system
Plan of Action and Milestones (POA&M), OMB Memorandum 02-01 [27]	Identifies tasks needing to be accomplished. Details resources required to accomplish the elements of the plan, any milestones in meeting the tasks, and scheduled completion dates for the milestones
Authorization Decision Document	Conveys the final security authorization decision from the authorizing official to the information system owner or common control provider, and other organizational officials, as appropriate. Contains the following information: (a) authorization decision, (b) terms and conditions for the authorization, and (c) authorization termination date. The security *authorization decision* indicates to the information system owner whether the system is: (a) authorized to operate; or (b) not authorized to operate

32.5 Supply Chain Protection

Supply chain risks are real; ground systems software and hardware need to be protected against them. NIST SP 800-53 Rev 4 has a system and services acquisition (SA) control with multiple enhancements, SA-12 Supply Chain Protection, to provide guidance in mitigating the threat from supply chain threats [2]. In addition, there is a NIST Interagency or Internal Report on the topic to provide more detailed guidance; "NIST IR 7622, Notional Supply Chain Risk Management Practices for Federal Information Systems" [28]. "This guide offers an array of supply chain assurance methods to help federal agencies manage the risks associated with purchasing and implementing information and communications technologies (ICT) products and services" [29]. The publication calls for procurement organizations to manage supply chain risk by using technical and programmatic mitigation techniques. The recommendations are based on information technology security practices and procedures expanded to include supply chain implications. The ten practices discovered, will if implemented in their entirety cover the system development lifecycle and include those listed here.

1. Uniquely identify ICT supply chain elements, processes, and actors
2. Limit access and exposure within the supply chain
3. Establish and maintain the provenance of elements, processes, tools, and data
4. Share information within strict limits

5. Perform supply chain risk management (SCRM) awareness and training
6. Use defensive design for systems, elements, and processes
7. Perform continuous integrator review
8. Strengthen delivery mechanisms
9. Assure sustainment activities and processes
10. Manage disposal and final disposition activities throughout the system or element lifecycle

Figure 32-3 depicts the essential elements of supply chain risk management practices.

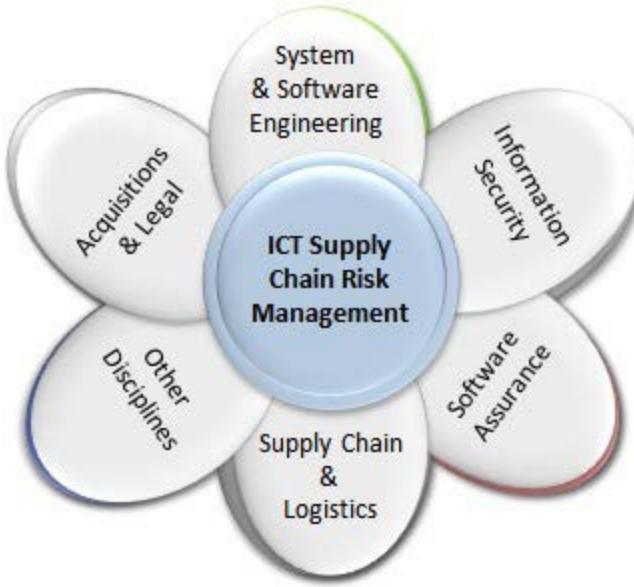


Figure 32-3. Components and contributing disciplines of ICT SCRM [28].

NIST SP 800-53 security control SA-12, Supply Chain Protection [2], states that "the organization protects against supply chain threats to the information system, system component, or information system service by employing [Assignment: organization-defined security safeguards] as part of a comprehensive, defense-in-breadth information security strategy." The Supplemental Guidance to this control stats that "Information systems (including system components that compose those systems) need to be protected throughout the system development lifecycle (i.e., during design, development, manufacturing, packaging, assembly, distribution, system integration, operations, maintenance, and retirement)." It states that protection should be

accomplished through threat awareness, by the identification, management, and reduction of vulnerabilities at each phase of the lifecycle and the use of complementary, mutually reinforcing strategies to respond to risk." Organizations should consider implementing a standardized process to address supply chain risk with respect to information systems and system components should be implemented, the acquisition workforce educated on threats, risk, and required security controls. The objective is to implement security safeguards that will (a) reduce the likelihood of unauthorized modifications at each stage in the supply chain; and (b) protect information systems and information system components, prior to delivery of such systems/components. There are twelve control enhancements to this control, SA-12 Supply Chain Protection, and include the following:

- Acquisition strategies/tools/methods
- Supplier reviews
- Limitation of harm
- Assessments prior to selection/acceptance/update
- Use of all-source intelligence
- Operations security
- Validate as genuine and not altered
- Penetration testing/analysis of elements, processes, and actors
- Inter-organizational agreements
- Critical information system components
- Identity and traceability
- Processes to address weaknesses or deficiencies

Of particular interest to software systems are control enhancements SA-12 Supply Chain Protection/Assessments Prior to Selection/Acceptance/Update, and SA-12 Supply Chain Protection/Penetration Testing/Analysis of Elements, Processes, and Actors. The Supplemental Guidance for the first, SA-12, states that assessments can include testing, evaluations, reviews, and analyses. Assessments of systems, components, products, tools, and services are conducted to uncover unintentional vulnerabilities and intentional vulnerabilities including malicious code, malicious processes, defective software, and counterfeits. Assessment techniques can include static analyses, dynamic analyses, simulations, white, gray, and black box testing, fuzz testing, and penetration testing for software components. The second enhance of particular relevance to software is SA-12[2]. The supplemental guidance for this states that it "addresses analysis and/or testing of the supply chain, not just delivered items. Supply chain elements are information technology products or product components that contain programmable logic and that are critically important to information system functions. Supply chain processes include, for example: (a) hardware, software, and firmware development processes; (b) shipping/handling procedures; (c) personnel and physical security programs; (d) configuration

management tools/measures to maintain provenance; or (e) any other programs, processes, or procedures associated with the production/distribution of supply chain elements. Supply chain actors are individuals with specific roles and responsibilities in the supply chain. The evidence generated during analyses and testing of supply chain elements, processes, and actors is documented and used to inform organizational risk management activities and decisions."

NIST SP 800-53 control SA-12, Supply Chain Protection, also references three other controls (with enhancements) that are useful in mitigating supply chain risk. These are SA-4 Acquisition Process, SA-11, Developer Security Testing and Evaluation, and SA-15 Development Process, Standards, and Tools.

SA-04 Acquisition Process includes the following requirements, descriptions, and criteria, explicitly or by reference, in the acquisition contract for the information system, system component, or information system service in accordance with applicable federal laws, executive orders, directives, policies, regulations, standards, guidelines, and organizational mission/business needs: (a) Security functional requirements; (b) Security strength requirements; (c) Security assurance requirements; (d) Security-related documentation requirements; (e) Requirements for protecting security-related documentation; (f) Description of the information system development environment and environment in which the system is intended to operate; and (g) Acceptance criteria.

SA-11 Developer Security Testing and Evaluation requires the developer of the information system, system component, or information system service to: (a) Create and implement a security assessment plan; (b) Perform [Selection (one or more): unit; integration; system; regression] testing/evaluation; (c) Produce evidence of the execution of the security assessment plan and the results of the security testing/evaluation; (d) Implement a verifiable flaw remediation process; and (e) Correct flaws identified during security testing/evaluation.

SA-15 Development Process, Standards, and Tools includes the following: (a) Requires the developer of the information system, system component, or information system service to follow a documented development process that: (1) Explicitly addresses security requirements; (2) Identifies the standards and tools used in the development process; (3) Documents the specific tool options and tool configurations used in the development process; and (4) Documents, manages, and ensures the integrity of changes to the process and/or tools used in development; and (b) Reviews the development process, standards, tools, and tool options/configurations to determine if the process, standards, tools, and tool options/configurations selected and employed can satisfy.

32.6 Summary

The most important concept to be gleaned from this document is that cybersecurity requirements must be addressed in the acquisition development process and not just in the assessment and authorization process. Cybersecurity affects all levels of the open systems interconnection model from the application down through the operating system to the physical implementation of the network infrastructure [30]. Cybersecurity requirements and security engineering must be an integral part of the acquisition of an NSS ground system to ensure that the security profile is adequate to protect the contained data, information, and capabilities.

32.7 References

1. NIST Special Publication 800-37 Revision 1, "Guide for Applying the Risk Management Framework to Federal Information Systems - A Security Life Cycle Approach," February 2010 INCLUDES UPDATES AS OF 06-05-2014: PAGE IX.

2. NIST Special Publication 800-53 Revision 4 "Security and Privacy Controls for Federal Information Systems and Organizations." April 2013.

3. Department of Defense Instruction 8510.01, Risk Management Framework (RMF) for DOD Information Technology (IT). March 12, 2014.

4. Federal Information Security Management Act 2002. Subchapter 111, Chapter 35, Title 44, United States Code.

5. Committee on National Security Systems Instruction 1253, Security Categorization and Control Selection for National Security Systems. March 15, 2012.

6. NIST Special Publications 800-53A, *Guide for Assessing the Security Controls in Federal Information Systems and Organizations*. Building Effective Security Assessment Plans. June 2010.

7. DOD Instruction 8500.01, Cybersecurity, March 14, 2014.

8. DOD Instruction 8000.01, Management of the Department of Defense Information Enterprise, February 10, 2009.

9. DOD Directive 8500.01E, Information Assurance, April 23, 2007.

10. National Security Presidential Directive-54/Homeland Security Presidential Directive 23, January 23, 2008.

11. CNSSI No. 1253 Security Categorization and Control Selection for National Security Systems, Version 215, March 2012.

12. DOD Instruction 5000.02. Operation of the Defense Acquisition System, January 7, 2015.

13. DOD Instruction 5200.39. Critical Program Information (CPI) Identification and Protection Within Research, Development, Test, and Evaluation (RDT&E), May 28, 2015.

14. CNSSP No. 12. National Information Assurance Policy for Space Systems Used to Support National Security Mission. Committee on National Security Systems, March 20, 2007.

15. Intelligence Community Directive (ICD) 503. Information Technology Systems Security Risk Management, Certification and Accreditation. September 15, 2008.

16. OMB Cirular, A-130, Appendix III.

17. CNSS Instruction 4009. National Information Assurance (IA) Glossary. Committee on National Security Systems. April 26, 2010.

18. Federal Information Processing Standards Publication 200. Minimum Security Requirements for Federal Information Systems. NIST, March 2006.

19. 44 U.S. Code Section 3542.

20. Office of Management and Budget. Circular A-130, Appendix III.

21. NIST SP 800-64 Revision 2 Security Considerations in the System Development Life Cycle October 2008

22. Panock, Marybeth and Meredith Hennan. *Secure Software Assurance Coding Guidance*, TOR-2013-00742, The Aerospace Corporation, El Segundo, CA. 2013.

23. Defense Information Assurance (DISA) Application Security and Development (STIG) and Checklist. 2011.

24. *Gartner Magic Quadrant for Application Security Testing*, G00246914. July 2, 2013.

25. Johnson-Roth, Gail A., Norman Y. Lao, Jessica Perry. *Mission Assurance Baseline (MAB) Version 2.5.5.* ATR-2013-00505, The Aerospace Corporation, El Segundo, CA. April 05, 2013.

26. NIST Special Publication 800-18, Revision 1. Information Security. February 2006.

27. Office of Management and Budget. Memoranda 02-01. Guidance for Preparing and Submitting Security Plans of Action and Milestones.

28. NIST IR 7622, Notional Supply Chain Risk Management Practices for Federal Information Systems

29. NIST Publishes Methods to Manage Risk in the Federal ICT Supply Chain From NIST Tech Beat: November 27, 2012 http://www.nist.gov/itl/csd/supply-112712.cfm

30. OSI model – Wikipedia https://en.wikipedia.org/OSI_model

32.8 Bibliography

GSAW Tutorial Cybersecurity Mission Assurance Baseline, March 18, 2013

SAFECode: Software Assurance: An Overview of Industry Best Practices http://www.safecode.org/publications/SAFECode_BestPractices0208.pdf

Microsoft Security Development Lifecycle Simplified Implementation of the Microsoft SDL, Updated November 4, 2010

OWASP Secure Coding Principles, https://www.owasp.org/index.php/Secure_Coding_Principles

OWASP Secure Coding Practices Quick Reference Guide, https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf

OWASP Top Ten, https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

CERT Secure Coding Standards, https://www.securecoding.cert.org/confluence/display/seccode/CERT+Secure+Coding+Standards

Common Weakness Enumeration – National Vulnerability Database, http://nvd.nist.gov/cwe.cfm

Building Security In Maturity Model BSIMM4, September 2012 http://bsimm.com/download/

Jerry Hoff, "Adding Security To Your Company's SDLC", presentation – Static Code Analysis Division Whitehat Security

Committee on National Security Systems Instruction No. 4009 National Information Assurance (IA) Glossary, Revised June 2006.

32.9 Acronyms

A&A	assessment and authorization
AC	access control
ACAT	acquisition category
AO	authorizing official
API	application program interface
ASDB	acquisition security database
AU	audit and accountability
BSIMM	building security in maturity model
C&A	certification & accreditation
CAPEC	common attack pattern enumeration and classification
CDRL	contract data requirement list
CERT	computer emergency readiness team
CM	configuration management
CNSS	Committee on National Security Systems
COTS	commercial off-the-shelf
CP	contingency planning
CVE	common vulnerabilities and exposures
CWE	common weakness enumeration
DAST	dynamic code analysis tool
DCAS	security design & configuration acquisition standards
DIACAP	DOD Information Assurance Certification and Accreditation Process
DOD	Department of Defense
DoDD	Department of Defense Directive
DoDI	Department of Defense Instruction
FIMSA	Federal Information Security Management Act
FOSS	free and open source software

FTP	file transfer protocol
GOTS	government off-the-shelf
HP	Hewlett Packard
HTTPS	hyper-text transfer protocol
I/O	input/output
IA	information assurance
IA	identification and authentication
IAST	interactive code analysis tool
IC	intelligence community
IT	information technology
IV&V	independent verification & validation
&/O	input/output
NDAA	National Defense Authorization Act
NDI	non developmental item
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
NSA	National Security Agency
NSS	national security space
ODNI	Office of the Director of National Intelligence
POA&M	plan of action and milestones
RFP	request for proposal
RMF	risk management framework
SAST	static code analysis tool
SC	system and communications protection
SDLC	software development lifecycle
SI	system and information integrity controls
SMC	Space and Missile Systems Center
SME	subject matter expert
SNAC	Systems and Network Attack Center
SSH	secure shell
SSL	secure socket layer
STIG	Secure Technical Implementation Guide
SWA	software assurance
T&E	test and evaluation
TFTP	trivial file transfer protocol

Chapter 33
Human Systems Integration

Brian E. Shaw and Richard S. Marken
Engineering and Integration Division
Space Systems Group

33.1 Introduction

Human systems integration (HSI) is a system-level discipline that is responsible for management of the planning and execution of human-related requirements and technical practices, and for establishing the interfaces between the various domain activities to ensure effective engineering/testing for mission and programmatic success. The HSI domains are: human factors engineering, personnel, habitability, manpower, training, safety and occupational health, and force protection and survivability (Figure 33-1). Safety and occupational health may be separated into three domains—safety, occupational health, and environment—to better reflect the independent technical communities. HSI is managed as part of systems engineering while each HSI domain is managed separately within the existing domain stakeholder community processes.

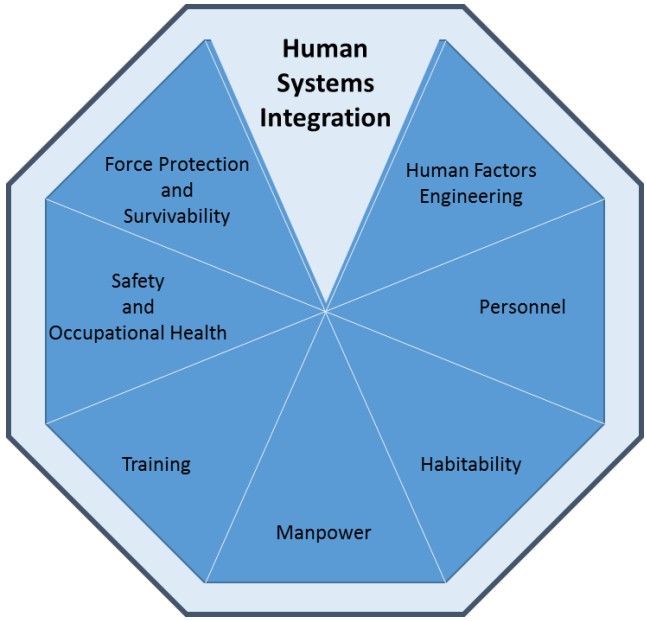


Figure 33-1. Human systems integration and domains.

The goal of HSI is to ensure continuity of the essential human-related requirements and implementation practices throughout the pre-acquisition, system acquisition, and sustainment phases of a system. This continuity helps assure that the system meets the required levels of system performance and balances technical considerations with cost and schedule constraints. HSI is an essential component of the systems engineering process, supporting total system cost management, affordability initiatives, and interoperability goals for system design and operation. HSI efforts seek to ensure the safety and support of warfighters as well as performance parameters for mission success.

The original focus of HSI was instrumentation (knobs and dials), but has evolved to a system science integrating all areas of human involvement in systems. The first known mention of HSI in systems engineering can be traced to one of the earliest systems engineering texts, *System Engineering: Introduction to the Design of Large-Scale System*s. [1] At that time the integration of human considerations into system design and operation was dominated by human factors engineering and safety communities with an emphasis on total system performance. Supporting the concern for lifecycle cost management, other human-related domain areas formed over the years, particularly addressing operations and sustainment issues such as occupational health, survivability, personnel, and crewing.

Human-related considerations in system design and operation has long been a core concern of the US Department of Defense (DOD) and federal agencies, especially the Federal Aviation Administration (FAA), that develop/implement large scale systems. The integration of disparate stakeholder communities and development of standardized processes and tools started in the early 1990s with the DOD per-service implementations of HSI, most notably the Army's highly successful Manpower Personnel Integration (MANPRINT) program. In 2002, the DOD issued a formal HSI policy in DOD 5000.2-R and that has been consistently maintained up to the current DODI 5000.02, specifically Enclosure 7. [2] A formal DOD Office of Deputy Secretary of Defense for Systems Engineering HSI initiative supports those requirements. In the first decade of the 21st century the Air Force has significantly advanced HSI implementation with guidance and tools for application of HSI throughout the entire lifecycle, including HSI requirements in systems engineering standards. These Air Force processes (which include documents and tools), while specific to Air Force systems, comprise the latest and most comprehensive material on the application of HSI in system development. These processes can be readily generalized to the development of any system within a DOD-style acquisition environment.

33.2 Definitions

Habitability Factors of living and working conditions that are necessary to sustain the morale, safety, health, and comfort of the user population and contribute directly to personnel effectiveness and mission accomplishment.

Human Factors Engineering The integration of human capabilities and limitations (cognitive, physical, sensory, and team dynamic) into system design, development, modification, and evaluation in order to optimize human-machine performance for both operation and maintenance of a system. Human factors engineering is the application of human factors data, information, and design criteria to the design of tools, machines, systems, tasks, jobs, and environments for safe, comfortable, and effective human use.

Human Systems Integration (HSI) The systems engineering function that integrates human capabilities and limitations throughout the system lifecycle effectively and affordably. HSI coordinates the activities of the following human-related functional areas, referred to as domains: manpower, personnel, training, human factors engineering, environment, safety, occupational health, survivability, and habitability. some organizations combine the domains of environment, safety and occupational health into a single "ESOH" domain.

Manpower The number and mix of personnel (military, civilian, and contractor) authorized and available to train, operate, maintain, and support the system.

Occupational Health Occupational health factors address risk of injury, acute and/or chronic illness, or disability and related reduced job performance of personnel who operate, maintain, or support the system.

Personnel The human aptitudes, skills, knowledge, experience levels, and abilities required to operate, maintain, and support the system at the time it is fielded and throughout its remaining lifecycle.

Training The instruction and resources required to provide system personnel with requisite knowledge, skills, and abilities to properly operate, maintain, and support the system.

Safety Design, procedural, and operational characteristics that address the possibilities for accidents or mishaps to operators and which threaten the survival of the system.

Survivability Characteristics of a system that address risk of fratricide, detection, and the probability of being attacked; and factors that enable the crew

to withstand man-made or natural hostile environments without aborting the mission or suffering acute and/or chronic illness, disability, or death.

33.3 HSI and System Engineering

HSI is a strategy that are executed as part of the systems engineering effort and is described in the systems engineering standards. [3] Aerospace has developed HSI requirements language that, in conjunction with the systems engineering standard, provides contractual compliance requirements for the effective execution of HSI. [4] Historically the approach used in acquisition has been to specify performance requirements for individual HSI domains that apply to the system but not address the coordination aspect of HSI. The Aerospace documentation addresses this shortcoming by providing specific guidance on how to coordinate HSI activities across domains. Other documentation has been developed to support requirements tailoring within human factors engineering (HFE), the HSI domain that most impacts ground segment development.

The first step in carrying out an HSI program is the identification of the domains that are applicable to the current system development. HSI should facilitate inclusion of the domain communities that must participate in the system development to ensure that the appropriate domain requirements are in place. A documented plan should address the inclusion of domains into the system acquisition/ operation processes, and the effective communication/interaction between the relevant domains. Communication can occur at formal technical reviews but is best initiated as early as possible within project integrated product team (IPT) activities.

Communication across domains involves the exchange of data and information. Each domain may give information to some domains and receive information from others. Table 33-1 shows, notionally, the level of communication between domains in terms of the number of data items (specific documents that may be invoked on contract) that should be exchanged for effective system development/operation. Measures of the required level of communication between HSI domains are based on the assessment of subject matter experts. The color of the cells indicates the degree of interaction between row and column domains in terms of the total number of data items to be exchanged. Green cells indicate 0 documents exchanged – meaning that the domains operate independently of each other and there generally is little interaction required between them. Yellow cells indicate 1–20 documents exchanged—a moderate level of interaction between domains. Red cells indicate greater than 20 documents exchanged—an intense level of interaction between domains.

The greatest amount of communication in terms of data item exchange is between the HSI systems engineering function (HSI/SE) and the individual HSI domains reflecting the role of HSI/SE as manager of the interactions between

HSI domains. The greatest level of communication is between HSI/SE and the HFE and safety domains. Otherwise there is little communication between the other HSI domains, showing the essential HSI role has for ensuring coordination between domains. The method used to derive this table and the specific data to be interchanged between domains is fully described in "Proposed Requirements for Human Systems Integration Planning". [5] The study was focused on DOD acquisition, however the results of the study could be tailored for a specific project, and easily genericized for non-DOD systems.

Table 33-1. The Degree of Interaction between HSI Domains, as Measured by Number of Data Documents to be Interchanged between Domains

	HFE	Manpower	Personnel	Training	Safety	Survivability	Occupational Health	Environment	Habitability
SE/HSI							0		0
HFE									0
Manpower			0	0	0	0	0	0	0
Personnel				0	0	0	0	0	0
Training					0	0	0	0	0
Safety						0	0	0	0
Survivability							0	0	0
Occupational Health								0	0
Environment									0

33.4 HSI Practices and Tasks

Four general HSI "tasks" are identified: planning, requirements, CONOPS/OPSCON, system analysis, testing planning, and testing. Tasks are also identified for the separate HSI domains and selected for associated engineering domains. Due to the overarching nature of the specialty engineering tasks, each HSI task is mapped as applicable to hardware, software and/or systems engineering, and applicable lifecycle phase.

Key HSI tasks for an acquisition program office include:

- Develop (and document) a program-specific plan to address HSI
- Define HSI requirements that includes compliance specifications, requirements, data items (products), and formal demonstrations and prototypes in the contractor
- Develop a team to address HSI issues (expertise in human factors engineering, systems engineering, software/hardware, logistics, and operations)

- Develop an effective approach to address transition and contractor interaction with operators early in the program

Key HSI tasks for contractors include:

- Develop a program-specific HSI plan in accordance with the specifications and requirements on contract
- Develop and execute the HSI domain activities in accordance with the HSI plan and the requirements on contract
- Develop a team to address HSI issues (expertise in human factors engineering, systems engineering, software and hardware, logistics, and operations)
- Execute HSI as part of overall systems engineering process (including SW, HW, etc.)

Table 33-2 provides a set of activities related to each HSI task area and each HSI domain. Development activities to be carried out are listed in the first column of the table. The next three columns indicate whether the activity is related to systems engineering (SE), hardware (HW) or software (SW) development. The last six columns indicate (with an X) whether the activity is to be carried out in a particular phase of the system development.

Table 33-2. Human Systems Integration Activities for Each HSI Domain during Each Phase of System Lifecycle Phase.

Human Systems Integration Activity	SE	HW	SW	JCIDS	Materiel Solution Analysis	Technology Development	Engineering and Manufacturing Development	Production and Deployment	Operations and Support
HSI Planning Develop HSI plan that identifies the appropriate HSI activities/products and applicable domains, including assurance that domain-level planning and intra-domain interfaces are accomplished	X	X	X	X	X	X	X		
Requirements Ensure that HSI requirements are included in programmatic and contractual artifacts to ensure conduct of an appropriate HSI program that includes all applicable domain-level activities	X	X	X	X	X	X	X		
CONOPS/OPSCON Assess the capability/system concept of operations from an HSI perspective to ensure that the operational	X			X	X	X	X		

Human Systems Integration Activity	SE	HW	SW	JCIDS	Materiel Solution Analysis	Technology Development	Engineering and Manufacturing Development	Production and Deployment	Operations and Support
concepts are complete, feasible and clearly stated relative to HSI domains									
System Analysis Assess if the functional analyses and trade studies have been performed to generate a functional architecture that supports safe and efficient operation and maintenance by the designated operators, maintainers, and users.	X	X	X	X	X	X	X		
Test Planning Assess the HSI test planning to ensure that HSI and domain requirements are integrated into developmental and operational testing for hardware, software, and system testing.	X	X	X		X	X	X	X	
HSI Testing Assess the human performance testing to ensure that the HSI integration and testing activities are complete, accurate, and fulfills the HSI and domain requirements	X	X	X			X	X	X	

Human Systems Integration Activity	SE	HW	SW	JCIDS	Materiel Solution Analysis	Technology Development	Engineering and Manufacturing Development	Production and Deployment	Operations and Support
HSI Domain-Level Activities									
Human Factors Engineering Ensure the design of the system such that the integration of human capabilities and limitations (cognitive, physical, sensory, and team dynamic) with the system design, development, modification and evaluation to optimize human-machine and human-system performance for operation, maintenance, sustainment and use of the system.	X	X	X	X	X	X	X	X	X
Personnel Ensure that the design supports the knowledge, skills, abilities, experience and aptitudes required to operate, maintain, and support the system at the time it is fielded and throughout its lifecycle.	X	X	X	X	X	X	X	X	X

Human Systems Integration Activity	SE	HW	SW	JCIDS	Materiel Solution Analysis	Technology Development	Engineering and Manufacturing Development	Production and Deployment	Operations and Support
Habitability Ensure that the design supports the factors of living and working conditions that are required to support the safety, health, comfort and morale of the user (operator, maintainer, and user) population.	X	X		X	X	X	X	X	X
Manpower Ensure that the system is supported by the number and mix of personnel (military, civilian, and contractor) authorized and available to train, operate, maintain, and support the system.		X	X	X	X	X	X	X	X
Training Ensure that the operators, maintainers, sustainers, and users of the system are provided with the required knowledge, skills, and abilities to operate, maintain, support and use the system.	X	X	X	X	X	X	X	X	X

Human Systems Integration Activity	SE	HW	SW	JCIDS	Materiel Solution Analysis	Technology Development	Engineering and Manufacturing Development	Production and Deployment	Operations and Support
<u>System Safety</u> Ensure that design and operational characteristics of the system minimize the probability for accidents or mishaps to the system or its operators, maintainers, sustainers, and users.	X	X	X	X	X	X	X	X	X
<u>Occupational Health</u> Ensure that the system minimizes the risk of injury, acute and/or chronic illness, disability and/or reduced job performance of personnel who operate, maintain, support, or use the system.	X	X	X	X	X	X	X	X	X
<u>Environment</u> Ensure that environmental factors for the operators, maintainers, sustainers, and users minimize degradation of performance and do not pose a safety threat.	X	X		X	X	X	X	X	X

Human Systems Integration Activity	SE	HW	SW	JCIDS	Materiel Solution Analysis	Technology Development	Engineering and Manufacturing Development	Production and Deployment	Operations and Support
Force Protection and Survivability Ensure that the system reduces the risk of fratricide, detection, and the probability of being attacked. Ensure that the crew is able to withstand man-made or natural hostile environments without aborting the mission or suffering acute and/or chronic illness, disability, or death.	X	X	X	X	X	X	X	X	X
Related Activities									
Maintainability Ensure that the design of the system (hardware and software) is designed to be maintained by the designated personnel with the specified skill levels.	X	X	X	X	X	X	X	X	X
Manufacturing Ensure the design of manufacturing environments is compatible with relevant requirements of each of the HSI domains.		X					X	X	

33.5 Acquisition Lifecycle

Throughout the system lifecycle, HSI involvement is important to ensure that HSI concerns are addressed early and at the level of detail necessary to ensure both mission and programmatic success. Each acquisition phase has HSI objectives that must be fulfilled to ensure that the subsequent phases are appropriately programmed to address concerns and activities. The objectives of each acquisition phase is built on the foundation of the prior phase.

33.5.1 Capability Needs and Requirements Development

In the DOD environment, capability and/or mission requirements originate in the Joint Capability Integration and Development System (JCIDS). During this initial "phase" of development capability and/or mission requirements for the system are established, including those for the human component as part of total system performance.

HSI addresses the broad human-related implications of the materiel and non-materiel alternatives that are considered during this phase. The following parameters, constraints, and considerations should be addressed: doctrine, organization, training, materiel, leadership and education, personnel, and facilities. These are addressed as part of a capabilities-based assessment and included in the analysis of alternatives (AOA) study plan.

33.5.2 Materiel Solution Analysis – Pre-Phase A

During this phase, potential materiel solutions to mission needs are assessed. The alternatives are documented in an AOA report and used to support the initiation of an acquisition process.

The HSI objective in Materiel and Solution Analysis (pre-Phase A) is to perform the trade studies and requirements development that address HSI concerns that arise during the initial technical review and alternative system review. The goals of each of these system reviews are as follows:

1. Initial Technical Review (ITR)
    - Review HSI in cost analysis requirements description documents.
    - Ensure HSI is included in the program's cost estimate with sufficient detail to support a valid program cost estimate
    - Provide HSI inputs for chosen materiel solution approaches, including assumptions, risks, and cost drivers
2. Alternative System Review
    - Ensure HSI considerations are addressed in the AOA for chosen solution and alternatives.

- Verify the system requirements are consistent with user needs and applicable domain standards

The outputs of this phase that must address HSI or applicable HSI domains are the:

- Draft system requirements
- Test and evaluation strategy
- System engineering plan (SEP)
- System safety analysis
- Support and maintenance concepts and technologies

Additionally, HSI inputs should be included in the draft capability development document (CDD), AOA, technology development strategy (TDS), and cost/manpower estimates

33.5.3 Technology Development – Phase A

This phase seeks to reduce technology risk, determine the appropriate technologies to be integrated into the full system, and to demonstrate critical technologies. The HSI-specific goals are: identify and evaluate critical HSI technology; update system-level HSI criteria and ensure traceability to defined system capabilities/constraints; address HSI risk/cost concerns; and ensure that applicable HSI elements are in the system specifications/plans.

The primary HSI objectives during technology development (Phase A) are to perform the trade studies and requirements development that address HSI concerns prior to systems requirements review, system functional review, and preliminary design review. The goals of each of these system reviews are as follows:

1. System Requirement Review
   - Validate HSI requirements are in system performance specification.
   - Ensure HSI performance requirements in CDD are testable and defined in system functional baseline.
   - Ensure HSI risks are in comprehensive risk assessment.
2. System Functional Review
   - Address HSI requirements in the system functional baseline and with lower-level performance requirements.
   - Ensure that program documentation and lifecycle management plan (LCMP) address HSI requirements, metrics, and development efforts.

- Ensure sufficiently detailed system requirements and functional baseline to support cost estimation.

3. Preliminary Design Review (PDR)
   - Design/develop/demonstrate enabling/critical system concepts and technology components.
   - Ensure preliminary design includes all domain-specific performance requirements.
   - Ensure overall system design includes appropriate HSI design factors.
   - Ensure that HSI risks are identified and managed.
   - Compete all safety-critical drawings.
   - Include HSI requirements, metrics and development efforts in program documentation and LCMP.
   - Evaluate the preliminary design for HSI risks, design shortfalls, and undocumented requirements.

The outputs of this phase that should address HSI are as follows:

- Preliminary design review report
- Test and evaluation master plan (TEMP)
- Systems engineering plan (SEP)
- Programmatic environmental safety and health evaluation (PESHE)
- Program protection plan (PPP)
- Technology readiness assessment
- National Environmental Policy Act compliance schedule
- Risk assessment
- Validated system support and maintenance objectives and requirements

Additionally, HSI inputs should be included in the integrated baseline review, the integrated support plan, system threat assessment, CDD, acquisition strategy, affordability assessment, and cost/manpower estimates.

33.5.4 Engineering and Manufacturing Development – Phase B

The purpose of this phase is to develop a system, capability, or increment of a capability that ensures full system integration as well as design for manufacturing/production/support and affordability. The HSI-specific goals are to participate in HSI-critical trade studies, coordinate trade-offs between individual HSI domains, coordinate with systems engineers on HSI trade-offs in system/sub-system/component requirements, and ensure that HSI concerns are properly integrated into the final design.

The HSI objective during the Engineering and Manufacturing Development phase (Phase B) is to participate in the design process and address HSI concerns before the delta preliminary design review, critical design review, test readiness review, system verification review, production readiness review, and functional configuration audit. The goals of each of these system reviews are as follows:

1. Delta-Preliminary Design Review
   - Ensure that open HSI issues/risks from Phase A PDR are in PDR assessment report
   - Review documentation for domain-specific requirements, analysis/decisions, and tasking
2. Critical Design Review
   - Ensure that HSI risks are identified, included in risk assessment, and are being managed
   - Verify that HSI design considerations are responsive to mission requirements in the CDD
   - Ensure that program documentation and LCMP includes HSI requirements, metrics, and development efforts
   - Ensure that HSI requirements and risks are addressed in design and software product specifications
3. Test Readiness Review
   - Ensure that HSI domain requirements are included in planned testing and procedures
   - Verify completeness of HSI risk documentation and acceptability by leadership
4. System Verification Review
   - Ensure system functionality documented in functional baseline fulfills HSI domain requirements
   - Ensure HSI risks are being adequately managed
5. Production Readiness Review
   - Ensure that HSI issues are addressed in production readiness and manufacturing processes
6. Functional Configuration Audit
   - Confirm the achievement of HSI functional and performance requirements

The outputs of this phase that must address HSI are:

- Initial product baseline
- Test reports
- Updates to the draft document outputs from phase A (TEMP, PESHE, PPP, SEP)
- Lifecycle sustainment plan
- System safety analysis

Additionally, HSI inputs should be included in the capability production document (CPD), STA, ISP, and cost/manpower estimate.

33.5.5 Production and Deployment Phase – Phase C

The purpose of this phase is to achieve the operational capability fulfilling a mission need and to verify fulfillment by operational testing. The HSI goal during this phase is to analyze and correct HSI domain-related deficiencies.

The HSI objective during a production and deployment phase (phase C) is to address all HSI concerns before the physical configuration audit. The goals of this audit are as follows:

1. Physical configuration audit
    - Ensure that HSI concerns in the system are accounted for by testing, measuring, and controlling
    - Ensure that HSI aspects of the as-built configuration are properly reflected in the procured data package
    - Identify all hazardous materials and processes in the technical data package

The outputs of this phase that must address HSI or applicable HSI domains are:

(a) Production baseline
(b) Test reports
(c) TEMP
(d) PESHE
(e) SEP
(f) System safety analysis

Additionally, HSI input should be included in the cost/manpower estimate.

33.5.6 Operations and Support Phase

The purpose of this phase is to support and sustain the system in a manner that meets operational and mission requirements and controls overall lifecycle cost. The HSI goal is to monitor, collect, and analyze operational data to determine root cause of HSI-related deficiencies and failures and implement appropriate corrective actions.

The HSI objective during the operations and support phase is to address all HSI concerns before the in-service review. The goals of this review are as follows:

In-service review

- Ensure that HSI considerations in risk, operational readiness, and technical status assessments are measureable
- Substantiate HSI-related assessments with support budget prioritization

Additionally, HSI input should be included in the

- CDD input for next increment
- Modifications and upgrades to fielded systems
- SEP
- System safety analysis

33.6 References

1. Goode, Harry H. and Robert E. Machol. *System Engineering: Introduction to the Design of Large Scale Systems*. New York: McGraw-Hill. 1957.
2. United States Department of Defense Instruction (DODI) 5000.02 (2015). *Operation of the Defense Acquisition System*. Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics. January 7, 2015.

3. IEEE 15288.1-2014. *IEEE Standard for Application of Systems Engineering on Defense Programs*. NY, NY: IEEE Standards Association, IEEE Computer Society; Software and Systems Engineering Standards Committee. May 13, 2015.

4. Shaw, Brian E. Tailoring of IEEE 15288.1 Specialty Engineering Supplement to TOR-2015-01949 The Aerospace Corporation. El Segundo, CA. [Also published by USAF Space and Missile Systems Center as SMC-T-006]. 2015.

5. Shaw, Brian E. and Richard E. Marken. *Proposed Requirements for Human Systems Integration Planning*. TOR-2012(8960)-1 REV A The Aerospace Corporation, El Segundo, CA. 2012.

33.7 Bibliography

There are many current and/or best-of-breed standards, handbooks, and data item descriptions (DIDs) supporting implementation of HSI. Many of these resources were used to develop the content of this chapter. Other government and commercial guides exist, as well as academic textbooks, but here are the most current documents that are appropriate to the implementation of HSI to space systems. Selected standards (but not handbooks or DIDs) for the HSI domain areas that impact space system development, operation, and sustainment

are also provided. Specific emphasis is placed on current authoritative documents that are generally applicable to the customers of The Aerospace Corporation. This list should not be considered all-inclusive. Contractor internal processes and proprietary command media for HSI and the associated domains are known to exist, but have not been evaluated and are not documented in this report.

33.7.1 Acquisition Policy with HSI Impacts

CJCSI 3170.01. Joint Capabilities Integration and Development System. 2015

DODI 5000.02. Operation of the Defense Acquisition System. 2015

33.7.2 Human Systems Integration

IEEE 15288.1. IEEE Standards for Application of Systems Engineering for Defense Applications. 2014

TOR-2015-01949. Specialty Engineering Supplement to IEEE 15288.1. El Segundo, CA: The Aerospace Corporation. 2015. [Also published by USAF Space and Missile Systems Center as SMC-T-006]

TOR-2012(8960)-1 REV A. Proposed Requirements for Human Systems Integration Planning. El Segundo, CA: The Aerospace Corporation. 2012

DI-HFAC-81743A. Human Systems Integration Program Plan. April 2007.

DI-HFAC-81833. Human Systems Integration Report. March 2011.

Defense Acquisition Guide. Chapter 6. Human Systems Integration. https://acc.dau.mil/CommunityBrowser.aspx?id=510172

Systems Engineering Body of Knowledge (SEBOK). http://sebokwiki.org/wiki/Human_Systems_Integration

INCOSE Systems Engineering Handbook. A Guide for System Cycle Processes and Activities, 4th Edition. International Council on Systems Engineering (INCOSE). July 2015.

Human Systems Integration (HIS) Requirements Pocket Guide. Air Force Human Systems Integration Office. September 2009.

Human Systems Integration (HSI) in Acquisition; Integrating Human Concerns into Life Cycle Systems Engineering; Management Guide. Air Force Human Systems Integration Office. August 2009.

Human Systems Integration (HSI) in Acquisition; Integrating Human Concerns into Life Cycle Systems Engineering; Acquisition Phase Guide. Air Force Human Systems Integration Office. August 2009.

HSI in Acquisition and Requirements. Version 1.0. Air Force Human Systems Integration Office. August 2009.

Systems Engineering Specialty Disciplines. Vol 2. USAF Space and Missile Systems Center. September 2011.

33.7.3 Selected Domains Applicable to Space Systems

33.7.3.1 All Domains

Human Systems Integration (HSI) in Acquisition; Integrating Human Concerns into Life Cycle Systems Engineering; Domain Guide. Air Force Human Systems Integration Office. August 2009

33.7.3.2 Human Factors Engineering Domain

MIL-STD-46855A. Human Engineering Requirements for Systems, Equipment, and Facilities. 2011

MIL-STD-1472G. Department of Defense Design Criteria Standard. Human Engineering. 2012

ANSI-HFES 100-2007. Human Factors Engineering of Computer Workstations. Santa Monica, CA: Human Factors and Ergonomics Society. 2007

TOR-2015-03599-volumes 1, 2, and 3. MIL-STD-1472F & MIL-STD-1472G Human Engineering Recommended Tailoring. El Segundo, CA: The Aerospace Corporation. 2015.

TOR-2015003671. Requirements Extraction and Tailoring for ANSI/HFES 100-2007: Human Factors Engineering of Workstations. El Segundo, CA: The Aerospace Corporation. 2015.

33.7.3.3 Safety Domain

MIL-STD-882E. *System Safety*. 2012

AFSPCMAN 91-710. *Range Safety User Requirements Manual.* Volumes 1-7. Air Force Space Command. July 2004.

33.7.3.4 Environment, Safety and Occupational Health (ESOH) Domain

NAS 411. National Aerospace Standard. Hazardous Material Management Program. Aerospace Industries Association. Arlington, VA. 2013.

NASA Standard 8719.14. Process for Limiting Orbital Debris.

TOR-2006(8583)-4474, Rev A. Requirements of End-of-Life Disposal of Satellites Operating at Geosynchronous Altitude. [Also published by USAF Space and Missile Systems Center as SMC-S-015]. August 2009.

TOR-2007(8506)-7154. Requirements of End-of-Life Disposal of Satellites Operating in Orbits with Perigees below 2000 Kilometers. [Also published by USAF Space and Missile Systems Center as SMC-S-022]. August 2009.

Environment, Safety, and Occupational Health (ESOH) in Acquisition; Integrating ESOH in to Systems Engineering. USAF Human Systems Integration Office. 2009.

33.8 Acronyms

AOA	analysis of alternatives
CDD	capability development document
CONOPS	concepts of operation
CPD	capability production document
DID	data item descriptions
DOD	Department of Defense
ESOH	environment, safety, and occupational health
FAA	Federal Aviation Administration
HFE	human factors engineering
HSI	human systems integration
HW	hardware
INCOSE	International Council of Systems Engineering
IPT	integrated product team
ISP	integrated support plan
ITR	initial technical review
JCIDS	joint capability integration and development system
LCMP	life cycle management plan
MANPRINT	Manpower Personnel Integration
OPSCON	operations concepts
PDR	preliminary design review

PESHE	programmatic environmental safety and health evaluation
PPP	program protective plan
SE	systems engineering
SEBoK	System Engineering Body of Knowledge
SEP	systems engineering plan
STA	system threat assessment
SW	software
TDS	technology development strategy
TEMP	test and evaluation plan
USAF	United States Air Force

Chapter 34
System Safety

Myron J. Hecht
Software Acquisition and Process Department
Software Engineering Subdivision

34.1 Introduction/Background

System safety is the application of engineering and management principles and techniques to achieve acceptable risk within the constraints of operational effectiveness, suitability, time, and cost throughout all phases of the system lifecycle. This chapter covers the system safety tasks and analysis techniques, including identifying hazards, assessing risk, defining risk mitigation measures, documenting residual and accepted risks, and using system safety processes as defined in MIL-STD-882E tailored to ground systems [1]. The objectives of system safety engineering are to ensure that:

- Safety, consistent with mission requirements, is designed into the system in a timely, cost-effective manner
- Hazards associated with systems, subsystems, or equipment are identified, documented, tracked, evaluated, and eliminated; or their associated risk is reduced to a level acceptable to the acquisition authority (AA)
- Actions taken are documented
- Required retrofit actions are minimized through the timely inclusion of safety design features during research, technology development, and acquisition of a system
- Changes in design, configuration, or mission requirements are accomplished in a manner that maintains a risk level acceptable to the AA
- Significant safety data are documented as lessons learned and are maintained
- Safety is maintained and ensured after the incorporation and verification of engineering change proposals (ECPs) and other system-related changes.

The system safety program encompasses all aspects of the development effort including

- Operational functions
- Maintenance and support functions
- Test activities

- Transportation and handling
- Operator and maintainer personnel safety
- Software/hardware interfaces
- Software/human interfaces
- Environmental health and safety

System safety principles include:

(a) Safety must be designed in. Critical reviews of the system design identify hazards that can be controlled by modifying the design. Modifications are most readily accepted during the early stages of design, development, and test. Previous design deficiencies can be exploited to prevent their recurrence.

(b) Inherent safety requires use of both engineering and management techniques to control the hazards of a system. A safety program must be planned and implemented so safety analyses are integrated with other factors that impact management decisions. Management activity must effectively control analytical and management techniques used to evaluate the system.

(c) Safety requirements must be consistent with other program or design requirements. The evolution of a system design is a series of tradeoffs among competing disciplines to optimize relative contributions. Safety competes with other disciplines; it does not override them.

The system safety group is responsible for the identification, tracking, elimination, and control of hazards or failure modes that exist in the design, development, test, and production of both hardware and software. This includes interfaces with the user, maintainer, and operational environment. The outputs of these efforts result in the identification of safety-critical functions that should be documented, tracked, and monitored for effectiveness as the design matures.

The software-specific products for safety engineering for ground systems may be defined in a separate software safety plan (SSP) or in a safety evaluation section of the software development plan (SDP). In either case, if system hazards can be mitigated by software development practices, then development documents and products need to be specified to support the system safety effort. These may include supplemental software requirements and design analyses, data flow diagrams (DFDs), functional flow analysis, software requirement specifications, and other specialized software safety analyses.

34.2 Definitions

The following definitions are taken from MIL-STD-882E [1], the primary safety standard used in SMC programs. These definitions are typical of those used in system safety in multiple industries.

Acceptable risk Risk that the appropriate acceptance authority is willing to accept without additional mitigation.

Acquisition authority (AA) The organization or agency responsible for acquiring the system

Causal factor One or several mechanisms that trigger the hazard that may result in a mishap.

Development authority (DA) The organization, agency, or contractor responsible for developing the system

Environmental impact An adverse change to the environment wholly or partially caused by the system or its use.

Event risk The risk associated with a hazard as it applies to a specified hardware/software configuration during an event. Typical events include developmental testing/operational testing (DT/OT), demonstrations, fielding, post-fielding tests.

Hazard A real or potential condition that could lead to an unplanned event or series of events (i.e. mishap) resulting in death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment.

Hazardous material (HAZMAT) Any item or substance that, due to its chemical, physical, toxicological, or biological nature, could cause harm to people, equipment, or the environment.

Human systems integration (HSI) The integrated and comprehensive analysis, design, assessment of requirements, concepts, and resources for system manpower, personnel, training, safety and occupational health, habitability, personnel survivability, and human factors engineering.

Initial risk The first assessment of the potential risk of an identified hazard. Initial risk establishes a fixed baseline for the hazard.

Level of rigor (LOR) A specification of the depth and breadth of software analysis and verification activities necessary to provide a sufficient level of

confidence that a safety-critical or safety-related software function will perform as required.

Mishap An event or series of events resulting in unintentional death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment. The term "mishap" includes negative environmental impacts from planned events.

Mitigation measure Action required to eliminate the hazard or when a hazard cannot be eliminated, reduce the associated risk by lessening the severity of the resulting mishap or lowering the likelihood that a mishap will occur.

Mode A designated system condition or status.

Probability An expression of the likelihood of occurrence of a mishap.

Risk A combination of the severity of the mishap and the probability that the mishap will occur.

Risk level The characterization of risk as either High, Serious, Medium, or Low.

Safety Freedom from conditions that can cause death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment.

Safety-critical A term applied to a condition, event, operation, process, or item whose mishap severity consequence is either Catastrophic or Critical.

Safety-critical function (SCF) A function whose failure to operate or incorrect operation will directly result in a mishap of either Catastrophic or Critical severity.

Safety-critical item (SCI) A hardware or software item that has been determined through analysis to potentially contribute to a hazard with Catastrophic or Critical mishap potential, or that may be implemented to mitigate a hazard with Catastrophic or Critical mishap potential. The definition of the term "safety-critical item" in this Standard is independent of the definition of the term "critical safety item" in public laws.

Safety-related A term applied to a condition, event, operation, process, or item whose mishap severity consequence is either Marginal or Negligible.

Safety-significant A term applied to a condition, event, operation, process, or item that is identified as either safety-critical or safety-related.

Severity The magnitude of potential consequences of a mishap to include: death, injury, occupational illness, damage to or loss of equipment or property, damage to the environment, or monetary loss.

Software system safety The application of system safety principles to software.

System The organization of hardware, software, material, facilities, personnel, data, and services needed to perform a designated function within a stated environment with specified results.

System safety The application of engineering and management principles, criteria, and techniques to achieve acceptable risk within the constraints of operational effectiveness and suitability, time, and cost throughout all phases of the system lifecycle.

System safety engineering An engineering discipline that employs specialized knowledge and skills in applying scientific and engineering principles, criteria, and techniques to identify hazards and then to eliminate the hazards, or reduce the associated risks when the hazards cannot be eliminated.

System safety management All plans and actions taken to identify hazards; assess and mitigate associated risks; and track, control, accept, and document risks encountered in the design, development, test, acquisition, use, and disposal of systems, subsystems, equipment, and infrastructure.

Target risk The projected risk level the PM plans to achieve by implementing mitigation measures consistent with the design.

34.3 Practices

The system safety activity should start at the definition of the mission need so that risks can be identified, considered in the Analysis of Alternative, and managed as part of the concept definition and formulation of the concept of operations. At the Phase A milestone, focus shifts from general concepts to the specific system under development to identify and mitigate safety risks until the residual risks are acceptable. Specific activities identified in MIL-STD-882E [1] and the *Joint System Safety Software Engineering Handbook* [2] include:

- Identify and document the safety-significant functions of the system in a functional and operational hazards assessment.
- Formulate and tailor generic or initial hardware/software (HW/SW) safety requirements or constraints to the system and software designers as early in the lifecycle as possible.

- Identify, document, and track system and subsystem-level hazards and their system-level effects. Categorize each identified hazard in terms of severity and probability of occurrence. Derive safety-specific hazard mitigation requirements to eliminate or reduce the likelihood of each causal factor.
- At Preliminary and Critical Design Reviews, review failure modes and effects analyses (FMEA) to identify each failure pathway and associated causal factors. Identify logical, practical, and cost-effective mitigation techniques and requirements for each failure pathway initiator.
- Provide engineering evidence that each mitigation safety requirement is implemented within the design, and the system functions as required to meet safety goals and objectives.
- Conduct a safety assessment of residual safety risk after all design, implementation, and test activities are complete.
- Conduct a safety impact analysis on all HW/SW change notices or ECPs for engineering baselines under configuration management.
- Submit for approval to the certifying authority all waivers and deviations where the system does not meet the safety requirements or certification criteria.
- Submit for approval to the acquiring authority an integrated system safety schedule that supports the program's engineering and programmatic milestones.
- Software-specific tasks can be integrated either into the system-level activities or included in the software development program. Software safety analysis should support the goals, objectives, and schedule of the parent system safety program.

34.3.1 Document the System Safety Approach

At program start, the system safety approach for managing hazards as an integral part of the SE process should be documented. The minimum content for the safety approach document include:

(a) The risk management effort and how it is integrated into the SE process, the integrated product and process development process, and the overall program management structure.
(b) The externally prescribed requirements are included in the system specifications and the flow-down of applicable requirements to subcontractors, vendors, and suppliers.
(c) Documenting hazards with a closed-loop hazard tracking system (HTS).
(d) Defining how hazards and associated risks are formally accepted by the appropriate risk acceptance authority and agreed to by the user in accordance with Department of Defense Instruction (DoDI) 5000.02 [3] .

34.3.2 Identify and Document Hazards

Hazards are identified by safety engineering analysis, are documented, and tracked in a closed-loop hazard tracking database system (HTS). This systematic analysis process includes hardware and software, interfaces, and the intended use in the operational environment. The data comes from: mishaps; environmental and occupational health; user physical characteristics; user knowledge, skills, and abilities; and lessons learned from legacy and similar systems. The hazard identification process considers the entire system lifecycle and potential impacts to personnel, infrastructure, defense systems, the public, and the environment. The HTS should include, as a minimum, the following data:

- identified hazards,
- associated mishaps,
- risk assessments [initial, target, event(s)],
- identified risk mitigation measures,
- hazard status,
- verification of risk reductions, and
- risk acceptances.

Both the AA and DA have access to the HTS. The AA ensures that the contract grants "government purpose rights" to all the data that the DA records in the database and any other items generated in the performance of the contract. Hazards should be maintained in a manner that they can be easily extracted and documented.

34.3.3 Assess and Document Risk

Risk is the product of occurrence probability and consequence severity. The system safety practice uses categories for both quantities. Severity category and probability levels should be assigned for each of the hazards described in the previous section across the system operating modes and states. Both probability levels and severity categories can be either generic or system specific.

Quantitative assessments of occurrence probability are possible where specific hazards or design basis events are known in detail and can be analyzed based on the physical properties of the system. When available, the use of representative quantitative data that defines frequency for the hazard is preferable to qualitative analysis. For software-intensive ground systems, a qualitative assessment method is often used. This is defined in MIL-STD-882E [1].

34.3.4 Identify and Implement Risk Mitigation Measures

The core actions of the system safety program is development of mitigations for risks identified in the risk assessment tasks and documentation of those mitigations in the HTS. The goal is to eliminate the hazard; when a hazard cannot be eliminated, the associated risk should be reduced to the lowest acceptable level within the constraints of cost, schedule, and performance. The following mitigation approach classes are listed in order of decreasing effectiveness.

(a) Eliminate hazards through design selection;
(b) Reduce risk through design alteration;
(c) Incorporate engineered features or devices;
(d) Provide warning devices; and
(e) Incorporate signage, procedures, training, and personal protective equipment (PPE).

The current status of hazards, their associated severity and probability assessments, and risk reduction efforts should be a part of all milestone technical reviews.

34.3.5 Verify, Validate, and Document Risk Reduction

The development activity should verify the implementation and validate the effectiveness of all selected risk mitigation measures through appropriate analysis, testing, demonstration, or inspection. The HTS should be used to document the verification and validation (V&V).

34.3.6 Accept Risk and Document

MIL-STD-882E requires that before exposing people, equipment, or the environment to known system-related hazards, the risks must be accepted by the appropriate authority as defined in DoDI 5000.02 [1, 3]. The system configuration and associated documentation that support the formal risk acceptance decision are provided to the government for retention through the life of the system. The user is part of the risk acceptance process and provides formal concurrence before all serious and high-risk acceptance decisions. After fielding, data from mishap reports, user feedback, and experience with similar systems or other sources may reveal new hazards or demonstrate that the risk for a known hazard is higher or lower than previously recognized.

34.3.7 Manage Lifecycle Risk

After the system is fielded, the program office uses the system safety process to identify hazards and maintain the HTS throughout the system's lifecycle. This

effort considers changes to include interfaces, users, hardware, software, mishap data, mission(s) and system health data. Risk management personnel, part of the configuration control process, are aware of these changes. If a new hazard is discovered or a known hazard is determined to have a higher risk level than previously assessed, the risk is formally accepted in accordance with DoDI 5000.02. In addition, the Department of Defense (DOD) requires program offices to support system-related Class A and B mishap investigations, as defined in DoDI 6055.07 [4], by providing analyses of hazards that contributed to the mishap and recommendations for materiel risk mitigation measures, especially those that minimize human errors.

34.4 Key Lessons Learned

There are many safety failures of ground systems that have resulted in significant losses of mission capability, including losses of entire missions. The causes of these failures has been

- Improper requirements, including omission of detection and mitigation of safety critical failures, incorrect or ambiguous requirements that resulted in safety mishaps, and requirements that resulted in incorrect or ambiguous displays to operators.
- Errors in system design including software design, interface specifications, and communications failures that resulted in unwanted actions resulting in failures.
- Incorrect assumptions on the behavior or capabilities of previously developed software [including commercial off-the-shelf (COTS)] that resulted in unanticipated and unwanted behavior.
- Incorrectly defined operator procedures that resulted in the wrong actions resulting in safety significant events and mishaps.
- Maintenance errors of omission and commission.

The table on the following pages was taken from failure and mishap records in The Aerospace Corporation's Space System Engineering Database (SSED) and was adopted from a presentation given at the 2012 Ground Systems Architecture Workshop (GSAW). It represents a small sample of the lessons that can be learned from ground system failures.

Table 34-1. Ground System Safety Significant Events Excerpted from the Aerospace SSED

Title	Event and Cause	Impact	Mitigation	Lesson Learned
No Command Checking (syntax)	Very egregious typographical error, supposed to be 5 places to left of decimal point and 7 to the right, decimal point left out in command. Software not designed to detect that type of error.	Vehicle outage	None	Command syntax and semantics checking requirements should have been allocated to the FSW or the ground system (or both)
No Command Checking (semantics)	An incorrect command load reversed configuration: Format 1 became High Gain, while Format 2 became Low Gain. Cause: TLM command semantics were slightly different based on which file was being processed	Payload data was lost for 15 days	The original gain settings were restored	Command semantics check on ground would have prevented this failure (in at least one location)
Insufficient Autonomy	(a) Ground station down due to antenna gain and output power issues (b) picosat malfunction upon deployment. Cause was battery exhaustion, a delay in ejection, or a minor over-temperature	Transmissions lost from most or all of the picosatellites.	Antenna issues resolved and operations returned after 13 days (too late for picosats)	Lack of autonomy functions in FSW combined with ground system unavailability caused loss of mission
Failure to Exit from Safe Mode	Phobos Grunt spacecraft rocket pack fails to fire	Simultaneous reboot of both SC computers resulting in safe mode from which no exit was possible; rocket firing could not occur	Vehicle lost	An alternate path is to enable ground restart of normal processing if no autonomous exit from safe mode would have saved the mission

SC = spacecraft, TLM = telemetry, FSW = flight software

34.5 Task Execution by Phase

The major milestones affecting the AA's safety and software safety program planning include the release of contract requests for proposals, proposal evaluation, major program milestones, system acceptance, certification testing, evaluation, production contract award, initial operational capability, and system upgrades or product improvements. The specific products to be produced by the accomplishment of the software safety engineering tasks are defined either within the System Safety Program Plan (SSPP) and the Software Development Plan (SDP). Figure 34-2 lists documents and analysis products aligned with program milestones defined in SMC-S-21. [5]

Table 34-2. System Safety Program Products

Document	1521B Program Milestone [4]
System safety program plan (SSPP)	Project initiation
Software safety program plan (SwSPP)	Projection initiation
Preliminary hazard list	System design review
Preliminary hazard analysis	System design review
Functional hazard analysis (FHA)	System design review (initial), preliminary design review (final)
Subsystem hazard analysis (SSHA)	Preliminary design review (initial) Critical design review (final)
System hazard analysis (SHA)	System acceptance
Operating hazard analysis (OHA)	Critical design review
Failure modes and effects analysis	Critical design review (updated throughout subsequent testing and development)
Safety case	Critical design review (updated throughout subsequent testing and development)

34.5.1 Pre-acquisition Planning

System safety begins when the need for the system is identified. The AA must incorporate the necessary language into the contract to ensure that the system under development will meet the safety acceptance or certification criterion. AA safety program planning continues through contract award and requires periodic updating during initial system development and as the development proceeds through its phases. Management of the overall system safety program continues through system delivery, acceptance, and throughout the system's lifecycle. After deployment, the AA must continue to track system hazards and risks and monitor the system in the field for safety concerns identified by the user. The

AA must also make provisions for safety program planning and management for any system upgrades, product improvements, maintenance, technology refreshment, etc.

34.5.1.1 Analyze and Comprehend the Conceptual Design Baseline of the System

The initial task involves gathering required data, information, and documentation for the initial assessment of the system. This assessment requires the system safety engineer to analyze the system and its intended operational environments to ensure that complete and accurate descriptions of the physical and functional aspects of the system are collected for initial safety analysis. Specific information to be reviewed and assessed includes, but is not limited to:

- Initial capabilities document (ICD) or capability development document (CDD)
- Concept of operations
- System specification
- Functional specification
- Software specifications
- Design engineering drawings (or presentation material)
- Operational view
- Interface specifications

In addition to the description of the system, it is important to comprehend the user requirements for the system and the operational environments in which the system will be deployed. These environments may be contributing factors in mishap and hazard scenarios that will be considered in the safety analyses.

34.5.1.2 Define Software Assurance Levels of Rigor (LOR)

The second step in the implementation of a software safety engineering program is to define the software assurance LOR tasks to be accomplished in the design, implementation, and test of safety-significant functions. This task begins with the identification of specific terms and definitions to be used for the program which must be formally documented in the SSPP and included in the SDP and Software Test Plan (STP). Specific definitions of safety-related, safety-critical, and safety-significant terms must be included in the SSPP.

34.5.2 Planning

The planning stage starts at the time of the formulation of the request for proposal (RFP) when the AA inserts contractual requirements and instructions into the RFP which will require offeror responses.

34.5.2.1 System Safety Program Plan (SSPP)

The SSPP is a requirement of MIL-STD-882E for Department of Defense procurements [1]. The SSPP details tasks and activities of the system safety engineering and management program established by the supplier. Specifically, the plan must include:

- A general description of the program
- The system safety organization
- System safety program milestones
- System safety program requirements
- Hazard analyses to be performed
- Requirements analysis to be performed
- Functional analysis to be performed
- Hazard analysis processes to be implemented
- Hazard analyses data to be obtained
- Method of safety verification
- Training requirements
- Applicable audit methods
- Mishap prevention, reporting, and investigation methods
- System safety interfaces

34.5.2.2 Software Safety Program Plan (SwSPP)

Because ground systems are software intensive, a requirement for SwSPP should be considered. The intent is to formally document that a development program is considering the processes, tasks, interfaces, and methods to ensure software safety. Specifically, to ensure that software safety is part of the design, code, test, and independent verification and validation (IV&V) activities to minimize, eliminate, or control the safety risk to a predetermined level. Software performing safety-critical functions should be assessed and controlled through sound system safety engineering and software engineering techniques. Elements of such a plan would include:

- Software safety requirements analysis
- Software safety design analysis
- Safety analysis of legacy code and externally developed software [including free and open source software (FOSS) and commercial off-the-shelf software (COTS)]
- Software safety code analysis
- Software safety test analysis
- Software safety change analysis

34.5.3 Hazard Analyses

Hazard analyses are used to identify potential conditions that could lead to an unplanned event or series of events (i.e. mishap) resulting in significant negative consequences. The results of hazards analyses are lists of hazards that are maintained in the HTS, which should include not only the hazard, but also the mitigation, whether through requirements definitions, design features, or the other mitigations discussed previously.

34.5.3.1 Preliminary Hazard List (PHL)

The purpose of a PHL is to compile hazards early in the development lifecycle, prior to the development of a functional architecture or detailed design. Therefore, the PHL must be generic. Sources of information that assist the analyst in compiling a preliminary list are:

- Similar systems hazard analysis,
- Historical mishap data,
- Lessons learned,
- Trade study results,
- Functional analysis,
- Preliminary requirements and specifications,
- Design requirements from design handbooks,
- Potential hazards identified by safety team brainstorming,
- Generic software safety requirements and guidelines, and
- Common sense.

Each hazard that is identified should contain

- The source of information
- Failure modes (i.e., the source or cause of the hazard),
- Mechanisms (often becoming the title of the hazard within the PHL), and
- Outcome (representing the potential mishap).

34.5.3.2 Preliminary Hazard Analysis (PHA)

The PHA activity is a safety engineering and software safety engineering function performed to identify the hazards and preliminary causal factors of the system under development. The hazards are formally documented to include the description of the hazard, causal factors, the effects of the hazard, and preliminary design requirements for hazard control by mitigating each cause. Performing the analysis includes assessing hazardous components, safety-significant interfaces between subsystems, environmental constraints, operation,

test and support activities, emergency procedures, test and support facilities, and safety-significant equipment and safeguards. The PHA format is defined in DI-SAFT-80101A [6] of MIL-STD-882E [1]. This DID also defines the format and contents of all hazard analysis reports. The PHA also provides an initial assessment of mishap severity and probability of occurrence. The software safety effort must assess and analyze the root causes. These causes should be separated into four categories:

- Hardware initiated causes,
- Software initiated causes,
- Human error initiated causes, and
- Human error causes that were influenced by software input to the user/operator.

The PHA becomes the input document and information for all other hazard analyses performed on the system including the subsystem and software hazard analyses.

34.5.3.3 Functional Hazards Analysis (FHA)

The FHA activity provides for the initial identification of safety-significant functions during the early design phase when the functional architecture has been defined. This activity must be performed as a part of the software safety process to ensure that the project manager, systems and design engineers, software developers, and test engineers are aware of each safety-significant or critical function of the design. This process also ensures the completeness of the FMEA and that each individual module of code that performs these functions is officially labeled as “safety significant” (either safety critical or safety related) and that defined levels of design and code analysis and test activity are mandated in the approved LOR table.

34.5.3.4 Subsystem and Software Hazard Analyses (SSHA)

The subsystem hazard analysis hazard analysis performed on individual subsystems of the (total) system. This analysis is launched from the individual hazard records of the PHA or FHA which were identified as a logically distinct portion of a subsystem. The SSHA is a more in-depth analysis of the functional relationships between components and equipment and software. This analysis addresses performance, performance degradation, functional failures, timing errors, design errors, or inadvertent functioning. Failure detection, failure isolation, failure annunciation, and control entity corrective action are also addressed. The SSHA analysis begins to provide the evidence of requirement implementation by matching hazard causal factors to design to prove or disprove hazard mitigation. The SSHA information includes, but is not limited to:

- hazard descriptions,
- all hazard causes
- hazard effects, and
- derived requirements to either eliminate or reduce the risk of the hazard by mitigating each causal factor.

The identification of a hazard cause can result in a derived requirement. The analysis should define preliminary requirements for safety warning or control systems, protective equipment, and procedures and training. The software safety analyses must define those hazards or failure modes that are specifically caused by erroneous, incomplete, or missing specifications, software inputs, or human error. These hazards are the basis for the derivation and identification of software requirements that eliminate or minimize the safety risk associated with the hazard. The SSHA must initiate resolution of how the system or subsystem will react if the software error occurs.

34.5.3.5 System Hazard Analysis (SHA)

The SHA provides documentary evidence of safety analyses of the subsystem interfaces and system functional, physical, and zonal requirements. As the SSHA identifies the specific and unique hazards of the subsystem, the SHA identifies those hazards introduced to the system by the interfaces between subsystems, man/machine interfaces, and HW/SW interfaces. The SHA assesses the entire system as a unit and evaluates the mishaps, hazards, failure modes, and causal factors that could be introduced through system physical and functional integration.

34.5.3.6 Operations Hazard Analysis (OHA)

The OHA identifies hazards during use of the system. It encompasses operating the system (primarily procedural aspects) and the support functions (maintenance, servicing, overhaul, facilities, equipment, training, etc.) that go along with operating the system. Its purpose is to evaluate the effectiveness of procedures in controlling those hazards, which were identified as being controlled, by procedures, instead of by design, and to ensure that procedures do not introduce new hazards.

OHAs are normally completed and submitted as a single package done in a matrix format. For a complex system, look for an analysis that is comprised of several separate analyses, such as for maintaining and servicing the system (sometimes called maintenance hazard analysis). This should go into the hazards of disconnecting and re-applying power, using access doors and panels, hardstands, etc. Past systems have had enough maintenance mishaps that a separate analysis is definitely justified.

34.5.4 Requirements Formulation

The requirements formulation phase has a critical effect on software safety. Most software errors in high-criticality systems can be traced to missing or misstated requirements [7, 8] and changing the software design or implementation in response to an error discovered during the requirements formulation phase can greatly increase the project cost for that change.

34.5.4.1 Requirements Affecting Software Safety

An important part of ensuring software safety during the requirements phase is to recognize and identify the unique requirements that space applications impose on software. Table 34-3 shows examples of such requirements.

Table 34-3. System Requirements Impacting Software Safety

Requirements Area	Impact
Power and weight constraints	Computing platform options
Functional requirements (TT&C, GN&C, payloads, life support, etc.)	Software size and complexity
System level response times	Computing system and software architecture and allocation within architecture
Autonomous failure detection and recovery and telemetry for Earth based failure and recovery	Architecture, software design, operational concepts
Constraints imposed by communication systems bit error rates, bandwidths, protocols	Software design and operational concepts
Mission duration and lifetime requirements	Computing platform options and software testing program
Failure probability	Software testing program
Difficult areas of functional requirements	Design and software

TT&C = telemetry, tracking, and command; GN&C = guidance navigational control

34.5.4.2 Requirements Verification

The primary product of system safety engineering analysis is the identification and communication of requirements to eliminate or reduce the safety risk associated with the design, manufacture, fabrication, test, operation, and support of the system. These requirements must be verified to be necessary, complete, correct, and testable for the system design. is it the responsibility of the system

safety function to identify, document, track, and trace hazard mitigation requirements to the design, and participate as such in the verification activities.

34.5.5 Architecture Definition

Software architectures are a significant part of the system architectures and there is significant overlap in the definitions. Software-related concerns can dominate those of the system architecture. It is essential there be simultaneously interaction between the two activities.

34.5.5.1 Architectural Evaluations and Tradeoffs Affecting Software Safety

Software architectures are developed with a variety of considerations to include the following issues:

- *Extent of redundancy:* A key top-level design question is the extent of redundancy needed for the mission. The most conservative approach is stated as fail operational/fail operational/fail-safe.
- *Distributed vs. centralized architectures:* selection of software architectures requires careful consideration of the relative vulnerabilities of alternatives and a properly performed tradeoff.
- *Extent of modularity:* Modularity is a desirable architectural attribute because it facilitates uncoupled development, integration of revised components, and utilization of previously developed components. Modularity can increase the number of interfaces which need to be maintained, and may introduce increased complexity and delays, increasing the likelihood of a failure.
- *Point-to-point vs. common communications infrastructure:* Whether the hardware communications structure utilizes common bus or shared network communications, inter-software process communications can either be point-to-point or utilize a common software communications mechanism.
- *COTS or reused vs. reused/modified vs. developed software:* There are safety benefits from the re-use of software from a relevant operational environments. An understanding of the differences, constraints, and tradeoffs are necessary.
- *Redundancy and Diversity:* Redundancy and diversity are key for increasing the safety of the software architecture.

34.5.5.2 Requirements Allocation and Traceability to Architecture

The HW/SW architectural development process must ensure traceability from all requirements to ensure that they have been implemented. This traceability is

performed every time the requirements are changed and applies to requirements derived from the system architecture software safety. Such requirements will emerge from such issues as:

- Power, weight, and volume derived constraints on processor throughput, memory capacity, storage, interfaces,
- Architectural constraints on choice of languages, operating systems, other aspects of run time environments,
- Time and data synchronization,
- Throughput, response time,
- Architectural definition of fault containment regions,
- Effect of the architecture on fault management strategies,
- Effect of architecture on ability of humans to intervene for diagnosis and recovery,
- Message passing and message error handling, and
- Impact of differences between on-vehicle and vehicle-to-Earth communications

34.5.5.3 Software Architecture Verification Issues

Software architecture verification should address the following issues:

- *Conformance to performance constraints:* Constraints include throughput and response time. Estimates of average processing, capacity requirements, and latencies for each step in the processing string should be determined to ensure that the architecture can feasibly meet the requirements.
- Sufficient capacity and addressing hardware resources (i.e., memory and interfaces): The software architecture should be able to access all of the system resources.
- *Probabilistic analysis:* Where possible, software architectures should be modeled together with the underlying hardware architecture.
- *Safety and hazard analyses:* Safety and hazards analyses at the architectural level should identify critical failure modes, single points of failure, mitigation techniques at the architectural level, and derivation of mitigation requirements at lower levels.

34.5.5.4 Acquisition and Management Issues

Management issues affected by software architecture that impact safety include:

- Impact of architecture on necessary developmental skills and planning to acquire those skills in the development work force (through a combination of training and hiring)

- The technological, cost, and schedule risks architecture
- The industrial and technology base and future refresh requirements with respect to languages, software communications, and HW/SW interfaces;
- Ensuring complete documentation and appropriate V&V artifacts exist and conform to notation requirements (e.g., UML 2.0)
- Configuration management and change management of the architecture
- Ensuring that the resources, tools, and expertise are available for software architecture verification (inspections, automated tools, design reviews, others)

34.5.6 Software design

This sections issues and topics affecting software safety during the software design phase(s), include software design issues related to safety, verification issues, and acquisition management issues.

34.5.6.1 Software design issues related to safety

Some of the issues that need to be considered by developers of software design with respect to safety:

- *Traceability:* Requirements should be traceable to the procedures, functions, and classes defined in the design. For functional oriented architectures, duplications in functions should be minimized. Decomposition of higher-level functions into lower-level functions should be complete.
- *Exception handling and other failure behaviors:* Exception handlers should consider all failure conditions defined in the requirements and in the safety analyses, and those likely to occur within the module or class itself. Exceptions should be handled as close to the locations in the code where they are generated.
- *Diagnostics capabilities:* Requirements imposed by the architecture to sense and report on failures in its environment, including response time anomalies, priority inversion, and resource contention.
- *Redundancy management:* The redundancy management constructs in the design should be consistent with those defined in the architecture.
- *Implementation language:* The implementation language and runtime environment should be capable of realizing the design. Of particular concern are language features that support exception handling, timing constraints, checkpointing and logging, and recovery.
- *Interfaces:* Interfaces among software modules should be completely defined and include not only arguments for the inputs and outputs of the function or object itself but also additional parameters for status,

error handling, and recovery. Interfaces should be designed "defensively", i.e., to minimize failure propagation (parameter validation prior to use, strong typing, exception handling when constraints are violated).
- *Class library definition and inheritance:* For object oriented architectures the definition of base and derived classes should be consistent and traceable to both the requirements and the architecture.
- *Compatibility with hardware and resource constraints:* The software allocated to each hardware should conform to memory, processor capacity, and interface constraints
- *COTS and Non-developmental runtime environments:* Existing software components and runtime environments should be configuration controlled, and well characterized as to resource requirements, safety, and failure behavior

34.5.6.2 Verification Issues

The following are some of the safety-specific verification issues arising from the software design stage of development:

- *Traceability:* Completeness of the traceability of higher-level and derived software requirements to the design of individual software modules.
- *Functionality:* Correctness of the transformation of software requirements to software functionality.
- *Interfaces:* Software interface consistency and correctness.
- *COTS and non-developmental software:* Suitability of the re-used software components by means of assessment of operational service history, the applicability of the allocated requirements to the published capabilities of the software, compatibility with other runtime software, and the ability of the COTS software to support mission-unique failure recovery and fault tolerance strategies.
- *Safety:* Verification that software component failure behavior, fault tolerance provisions, and diagnostic provisions are in conformance with safety analyses performed at the architecture and at the design level (FMEA, fault tree analysis [FTA], others).

34.5.6.3 Management Issues

Some of the software development management issues affected by software safety include:

- Conformance to design standards and design documentation standards

- Consistent use of automated tools (e.g., generators of UML, program design language [pdl], etc.), specifically including annotation
- Extent of software re-use (COTS and non-developmental software)
- Planning for SW technology re-use
- Verification techniques (inspection, peer reviews, design reviews)
- Software design configuration management — particularly if the software design activity is being performed by multiple organizations
- Propagation of changes to software design to previous developmental stages (i.e., architecture and requirements) and subsequent (implementation and test)

34.5.7 Coding

This subheading discusses issues and topics affecting software safety during the coding phase.

34.5.7.1 Coding and implementation issues related to safety

Many safety concerns are common with other concerns related to software quality, readability, and maintainability and are not repeated here. The following are specific concerns related to software safety that apply during the coding phase:

- *Use of "safe" subsets for safety or mission-critical functions*: Non-deterministic constructs such as dynamic binding and dynamic memory reclamation can make software unpredictable. Languages such as Ada, C, C++ and Java have safe subsets with more verbose source code which can reduce productivity, complicate software maintenance, and discourage reusability.
- *Selection of subroutine or class libraries, and runtime environments:* The runtime libraries and environmental components should conform to the constraints of the architecture and design and provide the necessary capabilities to support desired failure behavior — including:
  - Reliability, performance, throughput;
  - Failure response, detection and recovery; and
  - Diagnostics requirements.

- *Definition of suitable coding standards and conventions:* Coding standards and conventions can enhance safety by considering such issues as:
  - Disallowing dynamic memory allocation in safety critical systems,
  - "Defensive" coding practices for out-of-range inputs and response times,
  - Exception handler implementation,

- Coding to enhance testability and readability,
- Documentation to support verification,
- Interrupt versus deterministic timing loop processing for safety critical software,
- Policies on allowable interprocess communications mechanisms;
- Permitted use of dynamic binding;
- Policies on initialization of variables; and
- Limitations on levels of inheritance.

- *Coding tools, static analysis tools, and development environments:* Coding tools, static analyzers, and integrated development environments aide document generation, enforce coding standards, design traceability, etc. Proper use can reduce the likelihood of defect introduction and increase the likelihood of their removal once discovered.
- *Configuration management practices:* Software defects are found in all testing phases; changes will be made in individual units. Defect tracking and configuration management practices should prevent uncertainty in software configuration.

34.5.7.2 Software Testing

Software testing methods are generally classified into two main categories: "black box" and "white box" or "glass box" [9] Black box methods test the software by disregarding the software's internal structure and implementation. The test data, completion criteria, and procedures are developed without consideration of the internal structure of the software test item. Black box testing is used at all levels of testing, and is particularly applicable at higher levels of integration where the underlying components are no longer visible.

"White box" testing, accounts for the internal software structure, in the formulation of test cases and completion criteria. White box testing includes branch testing and path testing. Branch testing requires that each branch or condition in a program be tested at least once. Path testing tests every path through a program at least once. White box testing is conducted at the 'unit' level, and at the unit integration level. It is rarely conducted at the higher system integration levels.

The following are some specific safety concerns:

- *Policies and practices on unit test:* Unit testing occurs after a software unit is developed and is a key part of defect removal. A variety of decisions on the practice of unit testing must be made and enforced uniformly including:

- Extent of structural code coverage (statement, branch, path, conditions)
- Variable ranges (nominal, boundary, off-nominal, extreme)
- Functional vs. “negative” testing
- Categories of testing to which units will be subjected — including more intensive test program criteria for safety critical software
- Use of automated testing tools

34.5.7.3 Management and Acquisition Issues

The following management and acquisition issues should be considered as part of this phase:

- Defect tracking
- Selection and enforcement of coding standards
- Selection and use of automated tools
- Metrics collection and analysis of coding discrepancies
- Documentation standards and requirements
- Planning for SW technology re-use
- Verification techniques (inspection, peer reviews, design reviews)
- Configuration management
- Propagation of changes to higher and lower levels

34.5.8 Integration Testing

Integration testing starts when completed units are combining into software subsystems, and continues until the final installation of the executable software into the operational or flight hardware. The testing issues described in the previous section on unit testing also apply to integration testing.

34.5.8.1 Resource Constrains

A test effort should have specific, quantifiable goals so that definite completion criterion can be established. However, many current national security space (NSS) ground software applications are so complex, and run in such an interdependent environment, that complete testing can never be achieved. Common factors in deciding when to stop are:

- Deadlines and milestones
- Test budget depleted
- Number of test cases completed with a specific percentage passing
- Nominal operation tests all pass

These constraints mean that testing objectives must be prioritized. As it is rarely possible to test every aspect of an application, safety assessments are used to prioritize and are performed at a minimum of three levels of indenture; preliminary or functional hazard analysis, preliminary system safety analysis, and system safety analysis, to determine the safety impact of the software components. Components whose failure have high safety impacts receive the highest priority in test resources [10, 11]. Other considerations include:

- Areas of greatest complexity
- Sections developed in rush or panic mode
- Historically problematic areas
- Areas of concern to the developers
- Results of FMEA for software [12].

34.5.8.2 Test Case Generation

Strategies for generation of integration test cases that can be used to increase either error detection effectiveness or test coverage efficiency and include

- *Input equivalence classes* [13]*:* Partition-testing strategies that exercise the same code and for which only one representative case is necessary.
- *Error classes* [14]*:* Limiting the number of test cases for each class of failure behavior.
- *Use of inspection results* [15]: Using the distribution of inspection—detected defects to drive the distribution of test data
- *Coupling dependency metric* [16]: Using the amount of coupling to focus test cases — particularly if a significant amount of software changes have been made.
- *Failure driven testing* [14**Error! Reference source not found.**]: Concentrating test cases on areas of the software where an abnormally high number of failures have been observed.
- *Robust testing* [17]*:* Selection of test case input data using a design of experiments approach.

34.5.8.3 Automated Testing

Automated testing reduces the manual effort required during later stages of testing, and can provide more thorough testing, more complete data collection and analysis, and repeatability. Automated test suites compress time as it occurs without manual oversight.

34.5.8.4 Software Test Staff Qualifications

The execution of software testing has historically been problematic. "[The] tests themselves must be designed and tested—designed by a process no less rigorous and no less controlled than that used for code." [13]. One difficulty is that software development organizations rarely recognize that software testing should be treated as an independent engineering discipline, not a repository for failed or junior employees. A second problem is that many software engineers mistakenly believe software testing is simply debugging software [13]. A good test engineer has a "test to break" attitude, an ability to take the point of view of the customer, a strong desire for quality, and an attention to detail.

34.5.8.5 Testing of Distributed Software

The most prevalent architectures for ground systems are distributed systems in which "client" tasks often run on different computers than the "server" tasks on which they depend. Safety specific testing that should be considered as part of the testing program include:

- *Assessing the safety of the underlying hardware and software implemented communications mechanisms*: Such mechanisms include middleware, the enterprise service bus, and network protocol stacks. If the middleware is COTS, such testing can often be performed early and independently of the application under development. This testing can be used to characterize and gain confidence in the software implemented communications mechanisms — or conversely, determine if they are in fact unsuitable. In either case, risk can be reduced through such an approach.
- *Failure/recovery testing:* One of the main issues in distributed systems is ensuring dependability in the presence of failures. Testing objectives should include Assessment of system behavior when a…:
  - *Communications link failures:…* communications link is disabled to determine if the failover characteristics of the software implemented communications mechanisms are effective
  - *Communications link degradations: …*communications link is degraded to determine if the software system tolerates degradations, and what is the impact on throughput and response time?
  - *Server task failures: …*server function is disabled to see if the system senses that the server has gone down, and if there is a redundant copy, can it resume functionality and rapidly re-establish communications with clients

- *Client task failures: …*client task fails to see if another client task can be automatically initiated and can it re-establish contact with all server tasks

34.5.8.6 Common Testing Concerns

The following concerns, which have been previously discussed, also apply to integration testing:

- Test plans and procedures
- Requirements traceability
- Recording of test results
- Collecting operating time for reliability analysis
- Practices for collecting and logging complete instances
- Analysis of test anomalies
- Testing to verify assumptions in safety analyses
- Test tools and support equipment
- Regression testing and impact of incremental development
- Failure/recovery and diagnostic test procedures
- Special testing for safety demonstration
- Failure reviews

34.6 References

1. U.S. Department of Defense, *MIL-STD-882E: Department of Defense Standard Practice: System Safety*, Headquarters Air Force Materiel Command/SES (System Safety Office), Wright-Patterson Air Force Base, 2013.

2. Joint Software Systems Engineering Working Group, *Joint Software Systems Safety Engineering Handbook*, U.S. Department of Defense, Indian Head, MD, August, 2010.

3. Office of the Undersecretary of Defense (AT&L), *Department of Defense Instruction (DoDI) 5000.02, Operations of the Defense Acquisition System and the Defense Acquisition Guidebook*, 26 November 2013. http://www.acq.osd.mil/docs/DSD%205000.02_Memo+Doc.pdf. [Accessed September 28, 2014].

4. Department of Defense Instruction 6055.07 Mishap Notification, Investigation, Reporting, and Record Keeping, June 6, 2011.

5. U.S. Department of Defense, *Technical Reviews and Audits for Systems, Equipment, and Computer Software (Aerospace)*, U.S. Air Force, 1985.

6. DI-SAFT-80101B Data Item Description System Hazard Analysis Report (SSHA), July 31, 1995.

7. Leveson, Nancy. "The Role of Software in Recent Aerospace Accidents," *Proceedings of the 2001 International System Safety Conference,* Huntsville, AL, September 10-15, 2001.

8. Hayhurst, K. and C. M. Holloway. "Challenges in Software Aspects of Aerospace Systems," *26th Annual NASA Goddard Software Engineering Workshop,* Greenbelt, MA, November 27-29, 2001.

9. Howden, W.E. *Functional Programming Testing*, Technical Report, Dept. of Mathematics, University of Victoria, Victoria, B.C., Canada, DM 146 IR, August 1978.

10. SAE ARP 4754, Certification considerations for highly-integrated or complex aircraft systems, Systems Integration Requirements Task Group AS-1C, Avionics Systems Division (ASD), Society of Automotive Engineers, Inc. (SAE), September 1995.

11. MIL-STD-882C, "Safety System Program Requirements", January, 1993 (superseded by MIL STD 882D which removed requirements for software).

12. Lutz, R. R. and R. M. Woodhouse. Experience Report: Contributions of SFMEA to requirements analysis. Proceedings of ICRE '96, pp. 44-51.

13. Beizer, B. "Software Testing Techniques", 2nd edition, Van Nostrand Reinhold Co. New York, NY, USA, pp. 404-05. 1990.

14. Musa, J. *Software Reliability Measurement Application*. McGraw-Hill, 1990.

15. Harding , J. T. "Using Inspection Data to Forecast Test Defects" *Crosstalk,* May, 1998, available online at http://www.crosstalkonline.org/storage/issue-archives/1998/199805/199805-Harding.pdf last visited August 10, 2014.

16. Binkley, A. B. and S. R. Schach. Metrics for Predicting Run-Time Failures and Maintenance Effort: Four Case Studies", *Crosstalk* , May, 1998, available online at http://www.stsc.hill.af.mil/crosstalk/frames.asp?uri=1998/08/predicting.asp last visited January 19, 2005.

17. Phadke, M. and K. Phadke. “Utilizing Design of Experiments to Reduce IT System Testing Cost”, *CrossTalk*, November/December 2011 available from http://www.crosstalkonline.org/storage/issue-archives/2011/201111/201111-Phadke.pdf, last visited August 10, 2014.

34.7 Bibliography

D. Branum, “50th SW completes transition to new GPS control system,” 19 September 2007. [Online]. Available: http://www.afspc.af.mil/news/story.asp?storyID=123068750. [Accessed 29 September 2014].

Department of Defense Joint Staff, ““2000 CJCS Master Positioning, Navigation, And Timing Plan”, CJCSI 6130.01B 15 available at last visited October 15, 2007,” June 2000. [Online]. Available: http://www.dtic.mil/doctrine/jel/cjcsd/cjcsi/6130_01b.pdf.

RTCA Subcommittee SC-159, “Integrity Failure Modes and Effects Analysis Aberration Characterization Sheets, RTCA Paper No. 034-01/SC159-867,” RTCA, Washington DC, 1997.

IBM Rational, “Overview of Rational DOORS,” IBM, [Online]. Available: http://pic.dhe.ibm.com/infocenter/doorshlp/v9r5/index.jsp?topic=%2Fcom.ibm.doors.requirements.doc%2Ftopics%2Fc_welcome.html. [Accessed 30 9 2014].

U.S. Department of Defense, “Department of Defense, MIL STD 1629A Procedures For Performing A Failure Mode, Effects And Criticality Analysis (through Notice 2),” http://assist.daps.dla.mil/quicksearch/, 1984.

P. Bishop and R. Bloomfield, “A Methodology for Safety Case Development: Perspectives on Safety Critical Sstems,” in *Proceddings of the Sixth Safety-Critical Systems Symposium*, 1998.

R. Weaver and T. Kelly, “The Goal Structuring Notation - A Safety Argument Notation.”, in *Proc. Of Dependable Systems and Networks Workshop on Assurance Cases*, http://www.aitcnet.org/AssuranceCases/agenda.html, 2004.

D. Manning, “Frequency Control,” *Frequency Control Symposium and Exposition,* vol. August, no. Issue , 29-31, p. 840 – 849, 2005.

R. N. C. S. a. P. J. Jeffrey Fedor, “Evolution of the Air Force Satellite Control Network,” *Crosslink, ,* vol. available from http://www.aero.org/publications/crosslink/spring2006/02.html, Spring, 2006.

34.8 Acronyms

AA	acquisition authority
CDD	capability development document
COTS	commercial off-the-shelf
DA	development authority
DFD	data flow diagrams
DID	data item description
DOD	Department of Defense
DoDI	Department of Defense instructions
DT	developmental testing
ECP	engineering change proposals
FHA	functional hazard analysis
FMEA	failure modes and effects analyses
FOSS	free and open source software
FSW	flight software
FTA	fault tree analysis
GN&C	guidance navigation and control
GSAW	ground systems architecture workshop
HAZMAT	hazardous material
HSI	human systems integration
HTS	hazard tracking system
HW	hardware
ICD	initial capabilities document
IV&V	independent verification & validation
LOR	level of rigor
NSS	national security space
OHA	operating hazard analysis
OT	operational testing
PDL	program design language
PHA	preliminary hazard analysis
PHL	preliminary hazard list
PM	project manager
PPE	personal protective equipment
RFP	request for proposal
SC	Spacecraft
SCF	safety critical function
SCI	safety critical item
SDP	software development plan
SE	systems engineering
SHA	system hazard analysis
SMC	Space and Missile Systems Center
SSED	space system engineering database

SSHA	subsystem hazard analysis
SSP	software safety plan
SSPP	system safety program plan
STP	software test plan
SW	software
SWSPP	software safety program plan
TLM	telemetry
TT&C	telemetry tracking and control
UML	unified modeling language
V&V	verification and validation