(1)

PROCEEDINGS

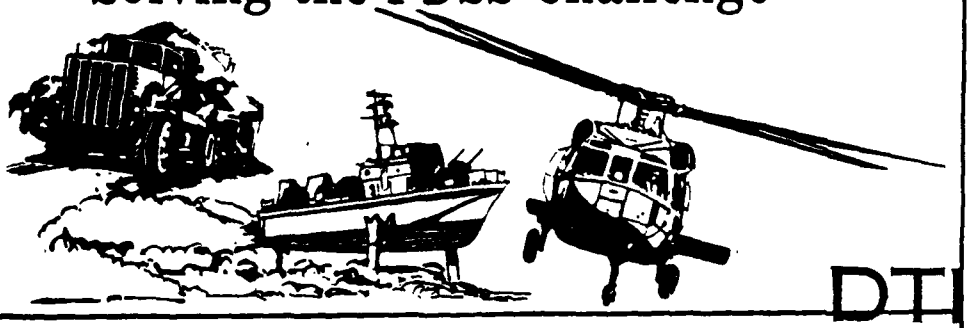
JUNE 1987

JOINT LOGISTICS COMMANDERS

4th Bienniel Software Workshop

ORLANDO II

Solving the PDSS Challenge



VOLUME II

This is not an approved JLC document.

JOINT LOGISTICS COMMANDERS 4TH BIENNIAL SOFTWARE WORKSHOP

"ORLANDO II"

VOLUME II

PROCEEDINGS

JUNE 1987

PRODUCED FOR THE
JOINT LOGISTICS COMMANDERS JOINT POLICY COORDINATING GROUP
FOR COMPUTER RESOURCE MANAGEMENT

THIS DOCUMENT WAS PRODUCED BY THE
POST DEPLOYMENT SOFTWARE SUPPORT SUBGROUP

Ms Shirley Peele, Chairperson	Navy (Surface)
Ms Lucille Cook	Navy (Air)
Mr Mel Weiner	Army
Capt Colin Gilyeat	Air Force (AFSC)
Mr Eldon Jensen	Air Force (AFLC)
Ms Myrna Olson	Marine Corps

Commander
FCDSSA
Dam Neck
Virginia Beach, VA 23461

FOREWORD

Orlando II is the fourth in a series of biennial workshops focusing on relevant software support issues pertinent to Mission Critical Computer Resources (MCCR). The previous workshops, Monterey I & II and Orlando I, were instrumental in the identification of issues that could be addressed in Department of Defense standards for the development of mission critical systems with embedded computers. The central theme of Orlando II was "Solving the PDSS Challenge". The workshop addressed all aspects of Post Deployment Software Support (PDSS) to identify areas which offer significant payoffs in terms of cost reduction, improved system reliability, and streamlining the PDSS process.

Specific panel topics were as follows:

I. PDSS Planning During Development.
II. Forecasting PDSS Resource Requirements
III. Software Change Process
IV. PDSS Standards
V. PDSS Management Indicators and Quality Metrics
VI. Human Resources in PDSS
VII. Software Technology Transition
VIII. MCCR Security

Orlando II addressed the problems causing the current crisis in support of MCCR. The workshop reinforced the fact that more cooperation is needed among the Services.

Volume I presents a summary of the issues and recommendations of the eight workshop panels. This Volume presents the workshop proceedings providing the details of each panels products and recommendations, and the guest speaker presentations. It is taken directly from the products provided by the panels without editorial comments, or reinterpretations.

Any questions concerning this material should be forwarded to:

Chairperson, PDSS Subgroup
Ms. Shirley Peele,
Fleet Combat Direction Systems Support Activity
Dam Neck, Code 82
Virginia Beach, VA 23461

Telephone: AUTOVON 433-7257 or (804) 433-7257.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

SECTION 1

INTRODUCTION

(Intentionally Blank)

INTRODUCTION

The Orlando II Workshop reviewed current Department of Defense (DOD) Post Deployment Software Support (PDSS) activities for Mission Critical Computer Resources (MCCR) and made specific recommendations to improve software support capabilities. Orlando II's purpose was to focus on the difficulties which have been experienced by both Government and industry agencies in support of software intensive systems and recommend solutions for those problems.

The central theme of Orlando II was "Solving the PDSS Challenge." Orlando II identified areas offering significant payoffs in terms of cost reduction, standardization of procedures, and improved system reliability, which will streamline the PDSS process. Secondarily, the workshop reviewed the status of recommendations made during the Orlando I Workshop, identified unresolved recommendations, and charted a course of action to complete any unfinished beneficial recommendations.

BACKGROUND

As a result of the growth of digital computer resources in weapon systems, it was necessary to standardize the development process of those systems. In 1977, the Joint Logistics Commanders (JLC) instituted the Joint Policy Coordinating Group for Computer Resource Management (JPCG-CRM) to accomplish this task. The mission of the JPCG-CRM is to:

> "...coordinate and ensure consistency in the preparation of new and revised regulations and standards, to provide recommendations on critical resource areas and to provide a focal point for coordinating standardization programs."

To accomplish their mission the JPCG-CRM have organized joint government/industry workshops. The workshops have been attended by experienced computer resource practitioners.

The first workshop, Monterey I, was held in 1979 at the Naval Post Graduate School at Monterey, California. Monterey I dealt primarily with software development and acquisition issues -- DOD policy, development standards, documentation standards, quality assurance standards and acceptance criteria. Two years later, at Monterey II, these issues were reviewed. New areas of concern were explored -- computer resource configuration item selection, standardization and accreditation of computer architectures, software cost estimating, and software reusability. These workshops identified the importance of coordination for support of MCCR.

The third biennial workshop, Orlando I, was held in late 1983. Monterey's I and II had focused on software development and acquisition. Orlando I focused on the support of MCCR after the initial development and deployment. The continuing and growing interest in the subject of post development and post deployment software support led the JPCG-CRM to form the PDSS Subgroup in June 1986. The PDSS Subgroup mission states:

> "The subgroup will identify, address, and resolve when possible, the problems and issues related to the maintenance and support phase of the life cycle."

One of the earliest requirements of the PDSS Subgroup was to:

> "...prepare and conduct an "Orlando II" workshop to revalidate or further definitize existing problems and define new ones requiring resolution."

The PDSS issues that were discussed at Orlando II are extremely important. They must be addressed now to meet future support requirements. Today PDSS efforts are consuming DOD resources at a devastating rate. Given the current budgetary environment, the Services cannot survive the projected growth in PDSS demands unless improved use and management of PDSS resources is attained. Actions must be taken to gain control over PDSS requirements. Operational capability must be maintained even while reducing the investment in already scarce PDSS resources.

PANEL GUIDELINES

PANEL I - PDSS PLANNINC DURING DEVELOPMENT.

Proper planning is necessary to enable efficient, effective software support after the developed software is deployed to the user. Software designs must consider the chosen support concept to facilitate the separation of software support responsibilities (e.g., Government, contractor, user). Software support tools, associated equipment, and facilities must be acquired in a timely fashion to permit the acceptance of support responsibilities by designated organizations. This has frequently not been done in the acquisition of MCCR software. Therefore, it is imperative that planning for support be performed during the development phase of MCCR software.

OBJECTIVE:

To identify those activities of MCCR software support that must be planned for during system development.

PANEL TASKS:

a. Identify, define, and prioritize PDSS activities that must be planned for during the software development phase.

b. Identify changes to current DOD regulations, standards, and directives to implement each aspect of planning identified above.

c. Identify methods of streamlining the budgeting process so necessary software support resources are provided at the time of system deployment.

PRODUCTS:

a. Prioritize list of PDSS planning activities.

b. Recommend specific modifications to DOD standards, directives, and regulations to implement each planning activity identified above.

c. Recommend improvements to the budgeting process.

PANEL II - FORECASTING PDSS RESOURCE REQUIREMENTS.

Successful planning for transition of new systems into operational use requires proper tools to forecast resource requirements. Accurate forecasting requires an in depth understanding of the system design, the selected support concept, interoperability issues, system support technologies, equipment, tools, and quantities and skills of personnel. Techniques must be developed to permit proper forecasting and budgeting for PDSS activities.

OBJECTIVE:

To identify a standard PDSS forecasting model.

PANEL TASKS:

a. Identify problems associated with current PDSS resource forecasting techniques.

b. Propose a standard PDSS forecasting model.

c. Identify and define the pertinent characteristics that must be included in the PDSS forecasting model.

d. Identify measurements that must be made to validate the PDSS forecasting model.

e. Identify areas that require further investigation and research to properly establish the model.

PRODUCTS:

a. A proposed standard PDSS forecasting model.

b. A comprehensive definition of characteristics and measurements included in the PDSS forecasting model.

c. Identification of further areas of investigation.

PANEL III - SOFTWARE CHANGE PROCESS.

The software support strategy decision requires an understanding of the functions that comprise software support, the software support process, and the advantages and disadvantages associated with each software support strategy alternative. Configuration management is a critical software support function which has the potential for significant cost avoidance if detailed implementing standards and common tools were to exist among industry and each Service. Existing DOD configuration management directives and standards must reflect the unique aspects of software configuration management not found in current hardware oriented configuration management documents. DOD configuration management methods must correctly reflect the unique nature of software configuration items.

SUBPANEL IIIA - PDSS MODELING/SUPPORT STRATEGIES.

OBJECTIVES:

a. To identify the functions involved in the software support process and to model that process.

b. To identify software support strategy alternatives.

SUBPANEL TASKS:

a. Develop a proposed joint Service software support model.

b. Develop a proposed joint Service software support contingency model which identifies those functions or activities that can be omitted or deferred to satisfy extraordinary user requirements (i.e., periods of national conflict, software faults that affect safety or mission capability, etc).

c. Identify those software support activities or functions that should be performed by government agencies vice contractual services.

d. Describe software support strategy alternatives and advantages/disadvantages of each strategy in terms of responsiveness, cost, risk, etc.

PRODUCTS:

a. A proposed joint Service software support model.

b. A proposed joint Service contingency model.

c. A proposed joint Service policy identifying software support activities or functions that should be performed within the government.

d. Description of software support strategies and the advantages/disadvantages of each.

SUBPANEL IIIB - CONFIGURATION MANAGEMENT.

OBJECTIVES:

a. To identify software and firmware related deficiencies in DOD configuration management (CM) directives and standards, and develop a recommended approach for implementing required changes.

b. Develop basic procurement documents for the development of an automated standard software configuration status accounting system.

SUBPANEL TASKS:

a. Review current DOD CM directives and standards and identify software and firmware related deficiencies.

b. Develop high level requirements outlining appropriate CM methods tailored to the unique nature of software and firmware configuration items.

c. Develop a recommended approach for incorporating the new requirements into existing DOD CM directives and standards.

d. Develop high level requirements and a Statement of Work (SOW) for a proposed automated standard software configuration status ccounting system.

PRODUCTS:

a. Report on software and firmware related deficiencies in current DOD CM directives and standards, including proposed new methods and recommendations for their implementation.

b. High level requirements, SOW, schedule, and cost estimates for the development of a proposed automated standard software configuration status accounting system.

PANEL IV - PDSS STANDARDS.

PDSS considerations are not adequately addressed in current software development standards. DOD-STD-2167 provides a thorough process for software development with DOD-STD-2168 complementing that process to ensure the production of quality software.

OBJECTIVE:

To identify changes to DOD software development standards to incorporate PDSS considerations.

PANEL TASKS:

a. Determine which requirements of DOD-STD-1467 should be incorporated in current software development standards.

b. Identify changes to DOD-STD-2167 needed to incorporate PDSS considerations.

c. Identify changes to draft DOD-STD-2168 needed to incorporate PDSS considerations.

PRODUCT:

Specific recommended changes to DOD-STD-2167 and draft DOD-STD-2168.

PANEL V - PDSS MANAGEMENT INDICATORS AND QUALITY METRICS.

Current management indicators and metrics focus mainly on the software development process. Measurements could provide valuable information to PDSS activities for planning and use during the maintenance process.

OBJECTIVES:

To identify management indicators and quality metrics applicable to PDSS.

PANEL TASKS:

a. Develop definitions and measurement criteria for software "ilities" as defined by DOD-STD-2167.

b. Identify a standard set of software metrics that provides a management assessment of software activities.

c. Identify a standard set of software metrics that provides technical assessment of software products.

PRODUCTS:

A set of standard DOD PDSS metrics for both management and technical assessment of PDSS activities and products.

PANEL VI - HUMAN RESOURCES IN PDSS.

Software support is a manpower intensive activity. Serious shortages in software personnel are impacting both industry and DOD software activities.

OBJECTIVE:

To define the actions necessary to ensure the acquiring, training, motivating, and retaining of knowledgeable software personnel in order to maintain a viable work force.

PANEL TASKS:

a. Identify issues in recruiting, training, and retaining MCCR software professionals.

b. Define the essential qualifications (academic and experience) required for MCCR software professionals, to include both practitioners and managers.

c. Identify means of enhancing individual productivity through training, incentives, and work assignments.

PRODUCTS:

Recommended solutions, or courses of actions necessary, to resolve the personnel related MCCR software support problems.

PANEL VII - SOFTWARE TECHNOLOGY TRANSITION.

PDSS activities consistently have received less than adequate software support tools from the developing agencies. Methods for the sharing of tools and new technology between the Services do not exist. Critical issues involve the acquisition of Ada* tools and software engineering environments.

*Ada is a Registered Trademark of the U.S. Department of Defense (Ada Joint Program Office)

OBJECTIVES:

To identify methods of transitioning necessary tools and controlling their proliferation so that PDSS needs are incorporated into the acquisition phase.

PANEL TASKS:

a. Identify problems and recommend solutions for the insertion of support and new technologies into PDSS activities.

b. Identify problems and recommend solutions for the transition of operational software (tactical programs) from the developing to the supporting organizations.

PRODUCTS:

a. Recommendations for improving the insertion of software support tools and new techniques to support activities.

b. Recommendations for improving the transition of operational programs from the developing to the supporting organizations.

c. Recommendations for controlling proliferation of support tools while not stifling new technology.

PANEL VIII - MCCR SECURITY.

Security accreditation is a costly and labor intensive effort. Current directives are incomplete, inconsistent, and do not adequately consider operational impacts with security requirement implementation.

OBJECTIVES:

To identify deficiencies with the DOD Security Program and recommend modifications to security regulations and industrial guidelines. (Note: The proceedings, conduct, and results are to be unclassified.)

PANEL TASKS:

a. Identify deficiencies with current industry and Service computer security regulations and guidelines.

b. Establish a list of multilevel security requirements necessary to support processing TOP SECRET, SECRET, and CONFIDENTIAL data within a single host computer. Local area and long haul network concerns should be addressed.

c. Map multilevel security requirements to the "ORANGE BOOK", OPNAVINST 5239.1A, and other Service and industry regulations and guidelines.

d. Identify where future research and development (R&D) should be focused.

PRODUCTS:

a. Specific recommended modifications to current industry and Service computer security regulations and guidelines.

b. List of recommended multilevel security requirements.

c. Comparison of security requirements to the ORANGE BOOK, OPNAVINST 5239.1A, and other Service and industry guidelines.

d. Prioritized list of areas that require further R&D.

(Intentionally Blank)

SECTION 2

GUEST SPEAKERS

(Intentionally Blank)

BIOGRAPHIES AND SPEECHES

The following pages contain the biographies of the Guest Speakers and the speeches they gave during the Orlando II Workshop. The biographies and speeches have been printed in the same order as they were scheduled to appear.

SCHEDULE OF GUEST SPEAKERS

Monday, 26 January 1987 Key Note Speaker	Lieutenant General Joseph J. Went, USMC, Deputy Chief of Staff for Installations and Logistics
Tuesday, 27 January 1987 Luncheon Speaker	Colonel Lewis E. Curtis, III, USAF, AFLC, Assistant Deputy Chief of Staff, Materiel Management
Wednesday, 28 January 1987 Luncheon Speaker	Brigadier General Alonzo E. Short, Jr., USA, Deputy Commander, Army Information Systems Engineering Command
Wednesday, 28 January 1987 Banquet Speaker	Rear Admiral Harry S. Quast, USN, Director, Information Systems Division
Thursday, 29 January 1987 Luncheon Speaker	Major General Monroe T. Smith, USAF, AFSC, Deputy Chief of Staff, Product Assurance and Acquisition Logistics

(Intentionally Blank)

LIEUTENANT GENERAL JOSEPH J. WENT

Lieutenant General Joseph J. Went is the Deputy Chief of Staff for Installations and Logistics, Headquarters Marine Corps, Washington D.C.

General Went was born in New Milford, Connecticut, on September 16, 1930. He received his Bachelor of Arts degree in Chemistry from the University of Connecticut in 1952. He also holds a Master's degree in Business Administration from George Washington University (1963), and graduated with distinction from the Naval War College, Newport, Rhode Island, in June 1972.

He entered the Marine Corps in July 1952 and was commissioned a Second Lieutenant in December 1952. Following completion of The Basic School, Quantico, Virginia, in June 1953, he underwent flight training at the Naval Air Station, Pensacola, Florida, and was designated a Naval Aviator in September 1954.

General Went has served with Marine attack, reconnaissance, transport, and fighter squadrons. He commanded Headquarters and Maintenance Squadron 12; Marine Attack Squadron 214; and Marine Aircraft Group 24. He has served in the Plans Division at Headquarters Marine Corps, and as the Comptroller of three commands: the Marine Corps Air Station, El Toro, California; Marine Corps Air Bases, Western Area, El Toro; and the 1st Marine Aircraft Wing.

He assumed duty as Chief of Staff, 1st Marine Brigade, Hawaii, in April 1976. While serving in this capacity, he was selected in February 1978 for promotion to Brigadier General. He was advanced to that rank in March 1978, and assumed duty as Deputy Fiscal Director of the Marine Corps, on April 1, 1987. In June 1980, he was assigned duty as Commanding General, 3d Force Service Support Group (Rein), Fleet Marine Force (FMF), Pacific, Okinawa, Japan. He was advanced to Major General on June 2, 1982 and assigned duty as Commanding General, 1st Marine Aircraft Wing, FMF, Pacific, Okinawa. General Went served in this capacity until May 1983. In June 6, 1983, he was assigned duty as the Deputy Commander, FMF, Pacific, Camp H.M. Smith, Hawaii. During June 1984, General Went reported to Headquarters Marine Corps where he was assigned as Deputy Chief of Staff for Reserve Affairs.

On June 6, 1985, he became Deputy Chief of Staff for Installations and Logistics with the rank of Lieutenant General. Lieutenant General Went holds the Legion of Merit with Combat "V", and three awards of the Air Medal.

Lieutenant General Went is married to the former Millie Cavaretta of Portsmouth, New Hampshire. They have two children: Sandie, who is married to Dr. James W. Smith of Athens, Alabama; and Angela, who is married to Major Randy J. Wijas, USMC.

SPEECH BY LIEUTENANT GENERAL JOSEPH J. WENT, USMC
Deputy Chief of Staff for Installations and Logistics
Monday, 26 January 1987

Good Morning!

Somebody asked this morning if I'm glad to be here. Listen, I'm glad to be anywhere. Especially after yesterday. I left my house at 1320 with the snow falling so hard you couldn't see across the street. I left early because I thought the cab would take a long time to the airport. It got there in record time, pulling directly up to the loading dock because there was no traffic at the airport. I remarked to the cabby that there wasn't much activity for a Sunday afternoon. He said, "Anybody would be an idiot to travel on a day like this."

At departure time the dispatcher announced that the FAA was closing the field while they plowed the snow from the runway. No problem. I had a good book. When we were finally called to board we were told that we would board but not depart for about 45 minutes because we'd have to stop at maintenance to have the plane deiced. I could go on but I won't. Glad to be here? You bet I am!

What qualifies me to be your keynote speaker? I'm not sure. It's only been a short time that I've realized that the kind of chips you talk about aren't served at happy hour. I suspect that each of you have a greater in-depth understanding of the subtleties of post deployment software support than I.

Thus, this will be plain talk or a layman's view of what we are all about. I do have certain responsibilities that bring me in direct contact with your everyday concerns. As the Deputy Chief of Staff for Installations and Logistics I am responsible for fielding all equipment in the Marine Corps, and service and support of that equipment until it is replaced or retired. I also co-chair the Marine Corps tactical software management steering committee. From that perspective, let's examine the world in which we live.

- Clearly, software engineering is a discipline which is still in its early stages of maturity.
- Emerging systems are increasingly sophisticated.
- There is a tremendous shortfall in civilian and military software professionals.
- Software quality often times is spotty.
- Adequate standards in both the software industry and the DOD are lacking.
- There is tremendous growth in both the size and number of systems.
- Software has become the pacing item in most C^3I and advanced weapons systems.
- All of the smart weapons that we need today and in the future get their brains from software.
- DOD is the largest single consumer of embedded software in the country and (I suspect) in the world.
- Costs of software to DOD are skyrocketing. Probably over $10 billion, that's more than the Marine Corps' entire operating budget.
- Software is nearly invisible except to the experienced (like yourselves).

Thus, there is a lack of understanding at the executive level of how to plan for and manage PDSS. This may be one of our greatest handicaps.

There are many software concerns but there are also many virtues. Two of the more significant virtues are:

- That small software changes can result in quantum system improvements; and
- Existing hardware can be substantially enhanced with software changes. We simply need to know how to bring this about.

I've taken a look at the stated purpose of Orlando II, i.e., to review current DOD PDSS activities and to make specific recommendations relevent to sofware support issues. This suggests that you know what the issues are. I'll talk about that in a bit.

Your central theme is "Solving the PDSS Challenge." I think you could have chosen a central theme "Solving the PDSS Crisis." Why crisis instead of challenge? Challenge has a connotation of a calling, an opportunity - the idea that there is free choice in the matter. Challenges can be savored and nurtured. They can often be set aside for awhile as we deal with something else. Challenge in my view doesn't impart the urgency of your business because challenges can often be dealt with in a long term way. Crisis, on the other hand, implies action now or failure

if we do not act in the right way. Crisis calls for immediate as well as long term solutions.

I'm not suggesting that you change the theme of Orlando II but I do suggest that you all recognize the urgent need for effective recommendations relevant to software support issues. A few weeks ago while thinking about what I would say today, I wrote down what I consider the major PDSS issues facing us. I tried to do this not only from a manager's but also a customer's and user's point of view. Let me share those concerns with you and try to relate them to your various panel activities.

First, how do we separate software responsibilities?

- Who's in charge?
- Who's responsible for what?
- How do we separate contractor, government and user responsibilities?
- How can we be sure that contractors will be responsive to our needs?
- How can we be sure that as managers we are prepared?
- How can we be sure our ultimate user's will be adequately trained and capable to make optimum use of the systems we develop?

I'm truly pleased to see that Panel I will get at all of these issues by keying on "PDSS Planning During Development."

My second concern is: What will we need in the way of resources to move from engineering development and operational test models to reliable, operational systems we can use daily and rely on to carry out our missions? Panel II gets right at this matter in their task of Forecasting PDSS Resource Requirements.

My third concern is: How do we manage change in such a way that we can do it quickly, inexpensively, and responsively without degrading system reliability while enhancing system capability? That is, how do we manage a configuration change? I'm pleased that Panel III gets right at the heart of the Software Change Process.

Fourth, I'm concerned that we must develop standards which will slow down the random activity we're involved in and provide some discipline to our efforts, so that both contractors and DOD activities operate within acceptable boundaries. It appears to me that Panel IV with its focus on PDSS Standards will get at this concern.

Next, I am concerned about the fact that while we have good performance indicators for almost all of our hardware, we do not have adequate ways of measuring progress and performance and the quality of our PDSS efforts. Thus, I am most pleased to see Panel V dealing with PDSS Management Indicators and Quality Metrics.

The last issue I wrote down, but the one I believe will be toughest to solve, is: How do we find the right people, in sufficient numbers to solve our PDSS professional personnel needs? We often approach problem solving by saying lets get the right people and they will find our solution. But because of the shortage of trained professionals, we will most often have to hire people with the potential to do the job for us after we have adequately trained them. Panel VI, you have your work cut out for you in dealing with Human Resources in PDSS.

As I thought about this morning a few weeks ago, I did not come up with the sort of issues that Panels VII and VIII will deal with. That is undoubtedly a fault of my layman's view. But the subjects of Software Technology Transition and Mission Critical Computer Resource Security are equally as important as the issues that I have raised. Fortunately, it is PDSS professionals that have set up Orlando II and have thought of these important issues.

Earlier I said that the purpose of this workshop suggests that you know what the issues are. Clearly from a review of your panel objectives and tasks, you do know the issues. And from my perspective, you are the right mix of people to work on these issues. A good mix of industry and defense professionals. Earlier, I also mentioned that there is a lack of understanding at the executive level of how to develop and manage PDSS. While it is not called for as one of your "products", I would urge you to develop a succinct executive summary of your efforts here this week. I may read your "whole book", but my superiors will not. Give me something that my Chief of Staff and Commandant would read - a Kiplinger style Executive Summary that will:

- Alert them to the issues;
- Enlist their involvement;
- Solicit their support.

Now, let me get out of here so that you can get to work. As you approach your tasks, I would ask you to do it with a true sense of urgency and that you do your best to have a little empathy from the customer's and user's points of view. It will make your product better. Thank you for listening to me. You have my best wishes for a successful week.

COLONEL LEWIS E. CURTIS, III

Colonel Lewis E. Curtis, III is Assistant Deputy Chief of Staff, Materiel Management, Headquarters Air Force Logistics Command, Wright-Patterson Air Force Base, Ohio.

Colonel Curtis was born in 1941 in Biloxi, Mississippi. After completing high school, he enlisted in the Air Force in 1960 and served as a radar maintenance technician on the F-105D. In 1964, he completed the Airman Education and Commissioning Program, and graduated from the University of Wyoming with a Bachelor of Science degree in Mechanical Engineering. He was commissioned in December, 1964. In 1969, he earned a Master of Science degree in Mechanical Engineering from the Air Force Institute of Technology. Colonel Curtis completed Squadron Officer School in 1970 and graduated from the Royal Air Force Staff College in 1974. In 1984 he graduated from Air War College and earned a Master's degree in Business Administration from Troy State University.

After receiving his commission and completing technical training at Chanute Air Force Base, Illinois, Colonel Curtis served as a maintenance officer on U-2, DC-130, and CH-3C aircraft and other special reconnaissance systems at both Davis-Monthan Air Force Base, Arizona, and Bien Hoa Air Base, Republic of Vietnam. In 1969, he returned to Southeast Asia and served as a Maintenance Officer on F-4D, RF-4C, C-130, and AC-47 aircraft.

Assigned to Headquarters Military Airlift Command in 1970, he served there as Chief of the Systems Analysis Branch in the Office of the Deputy Chief of Staff, Logistics. His subsequent assignments included an exchange tour with the Royal Air Force, where he managed the F-4K and F-4M aircraft for Headquarters Strike Command (RAF); Commander of the 934th Organizational Maintenance Squadron, 1st Special Operations Wing, operating AC-130, MC-130, UH-1N and CH-3C aircraft; Director of Logistics for the Advanced Medium Range Air-to-Air Missile (AMRAAM); and Deputy Director of Logistics for the B-1B. Before assuming his present post, he served in the Office of the Deputy Chief of Staff for Materiel Management at Air Force Logistics Command headquarters as both Deputy Director of Acquisition Logistics Policy and as Director of Engineering.

Colonel Curtis' decorations include the Defense Meritorious Service Medal, the Meritorious Service Medal with one Oak Leaf Cluster and the Air Force Commendation Medal. He assumed his present duties in June 1986.

Colonel Curtis is married to the former Kathleen Taylor of Biloxi, Mississippi. They have two sons.

(Intentionally Blank)

SPEECH BY COLONEL LEWIS CURTIS, AFLC
Assistant Deputy Chief of Staff, Material Management
Tuesday, 27 January 1987

Colonel Curtis opened his remarks by relaying General McCoy's disappointment in not being able to attend. Col Curtis noted that Gen McCoy supported the belief that software maintenance is an appropriate maintenance activity. Rather than reading Gen McCoy's prepared script, Col Curtis stated that he had reviewed and considered Gen McCoy's script and would present what he felt was the general perspective of senior staff members regarding software support issues. The following is a transcription of Col Curtis' remarks:

Post Deployment Software Support is the core of the Air Force Logistics Command's software business. Software development in the Air Force is done by the Air Force Systems Command. Once the system is deployed we have a mechanism we call Program Management Responsibility Transfer, whereby the Air Force Logistics Command takes over the program management and engineering responsibility for the weapons system for the remainder of its life. Included

in that transfer is the responsibility for the post deployment support of all the computer resources.

To give you an idea of the magnitude of the effort of the command, we have about 2,500 organic resources dedicated to the support of mission critical computer systems. Those include aircraft, communications systems, etc., and those are outside of the Automatic Data Processing community. In addition, last year we had about 800 man-years worth of contractor support, much of it in house Operations and Maintenance funded, supporting embedded computer systems, and that excludes the money we have in budget program nineteen for electronic countermeasures, the money we have in specialized acquisition and support management for such systems as the SR-71 and those software support resources we fund out of R&M and Research and Development money that's flushed to us through the Systems Command. Net, we probably have over 4,000 man-years per year dedicated to the support of Mission Critical Computer Resources. In facilities we have over six hundred thousand square feet of computer floor space in our establishment. Like many of you I've talked to here, we support over a hundred different processors in our embedded computer systems, over a hundred languages and dialects, and I think we share in common with you in our support facilities a number of the common problems that you have been discussing for the last day and a half.

We have a number of initiatives in the Command to address our growing computer resource and workload. We see major growth areas particularly in the ground segment of space systems. As some of you know, the Air Force has established within the past few years a Space Command, which is taking over the operations of the Air Force's space assets, and the ground control segment of those are software intensive. They have been treated as Research and Development systems in the past and we are in the process of "normalizing" the software support for those systems, which has been a challenge. In addition, we have a very discrete program to bring new technologies into the software support environment. Rick Holsman, among his other hats, is the program manager for software technology within the Command. We have another program office working Artificial Intelligence. We have a number of other offices working various Very High Speed Integrated Circuit applications, Very High Speed Integrated Circuit insertions in weapons systems; in fact, at Warner Robbins we are working an F-15 Very High Speed Integrated Integrated Circuit central computer and intend to move the operational flight program for the F-15 to Ada once we have that computer available. That's an overview of the current involvement in PDSS.

I was very impressed yesterday morning by our keynote speaker when he made what I believe is an important point about this conference and its objective that what we face is not a challenge in PDSS but rather a crisis in the support of mission critical software. However, I believe that we will probably disagree -- in fact I've heard disagreement in some of the panel sessions -- about the nature of that crisis, and that's really what I want to talk to you about today.

I think that there are two basic perspectives on what the problems really are. One is the perspective of the software practitioners, which encompasses most of the people here. The issues that I have heard discussed in the panels are: too few people to really do the job that we see out there; too few qualified

people to accomplish the complex nature of the work; and poor tools, or the wrong tools, to do the work in a timely manner. A pervasive theme in most of the panels has been: A clumsy beauracracy that doesn't understand the nature of the software work and is not responsive to the software requirements.

There is too little money or, again, the wrong kinds and colors of money to do the work. Finally, there is a management structure (especially a senior structure) that doesn't understand the essential nature of the software problems. All these factors are very true and I feel that these problems are exactly what you would expect given the state of the maturity of software to date.

About six months ago I was up at the Software Engineering Institute for a meeting and somebody put it in perspective for me. They described the state of maturity of software engineering as being equivalent to the state of maturity of Civil Engineering before Pythagoras invented the right triangle, and I think there's a lot of validity in that. That is the source of a lot of the software practitioner's problems with both bureaucracy and with senior management.

I could commiserate with you on the problems, but, rather, I'm going to adopt a devil's advocate approach and I'm going to give you what I believe is the view from the senior management and the nonsoftware people who view software activity in their midst as basically a black hole that will suck up all the resources that can be poured into it with very little visible results. One of the reasons for that criticism is an inherent characteristic of software. One of Major Randy Adam's lieutenants described software as being like entropy: It doesn't weigh anything, it's very difficult to grasp, and it obeys the first law of thermodynamics -- it always increases. That's true and also very scary. Like entropy, understanding software is almost an art as opposed to a science. I'll give you two views of the software world. The first is the operator's view, and I get pounded with this view frequently when I go out to the operating commands. It is the view attributed to our senior combat commanders in the Air Force. That view is that today's software, or software intensive systems, offer a real benefit because they provide capabilities that we've never had before. For those who have seen the X-29 -- if it wasn't for software that airplane could not fly. It is so inherently unstable that the guys at Grumman said "It would disintegrate in less than a second at cruise airspeed if it wasn't for the continuous flight control corrections from the flight control system."

But, on the other side, the very nature of embedding all of these capabilities in a computer implies that we're burying into software a lot of the things that the pilot or the aircrew used to do. The combat commanders perceive that their ability -- and I'm going to look down because these are words that General Creech originally used and General Russ recently echoed -- that, "The combat commander's ability to execute operations is increasingly being controlled by technocrats."; that's you and me. He warns his own people that, "The combat commander must retain the capability to change the way he fights the war and not allow flexibility to be controlled by the people writing the software."

The one thing that we can be sure of about combat -- be it war in Europe or Southeast Asia or anywhere else -- is that we really don't know how it's going to go. Uncertainty is an essential element of war. Clausewitz calls it the "fog of war." We absolutely know that regardless what plans we lay for combat in Europe, they are going to be wrong -- we are not going to execute the war the way we plan it. But, increasingly, the way we intend to fight that war is buried in the software we write. An example is the F-111, which, you know, is a low altitude penetration aircraft with a very sophisticated terrain following radar system, and it is set up to penetrate at a set altitude, for example, say 200 feet. Two hundred feet minimum penetration altitude is based on the performance characteristics of the radar flight control system, and is high enough to keep it from running into the side of a hill reliably when you're penetrating. Now, 200 feet is a great minimum penetration altitude, but after the first four days of war in Europe, we may well decide that we are better off penetrating at a hundred feet and occasionally running into a hillside rather than taking the attrition that a 200 foot penetration would give us. But, going back and changing the algorithms in the radar and the OFP to allow a hundred foot penetration is not a trivial exercise. That's not something you are going to do by sending out new ops instructions to the aircrew. That's something that we, the technocrats, control.

Another example is in both the new F-15E and the B-52 equipped with cruise missiles and the B1. All of those have very complex mission profiles and the operating command has a mission data preparation system that lays out the entire flight path and the way that the airplane is going to fight the battle. If you look at a typical B-1 strategic sortie, it may involve two or three refuelings, a penetration over the Black Sea, a number of short range attack missiles being launched against close-in defenses, and that will launch over the Crimea of a number of cruise missiles that are targeted for areas that are as far away as the Kola Peninsula, Murmansk, or Central Asia. That's all great when you put data in this system and feed it in before flight. But in the number of hours it takes the aircraft to penetrate into the Crimea from Dyess Air Force Base, it's unlikely that the situation he finds when he gets there is going to be the one anticipated when the mission preparation data tape was actually built. Again, the "fog of war", and our software is increasingly taking away the flexibility the combat commander needs for fighting.

Then, I'll give you another view, and that's the view of the supporters, which is basically the role that I play. I'm going to talk about three different things. The first is the transition of the development process to the support process. We do that very formally in the Air Force Logistics Command, with the handoff from the developing command to the supporting command. But, even if the support remains with that prime contractor, you still have that intellectual transfer as the bright young men who developed the system move on to the next challenge and other people fill in behind them to do the routine system maintenance. I think the characteristics of that transfer are fairly consistent across all three Services, in my experience, and even on those systems that we leave with industry. The first is that we begin thinking about the transfer and planning for the transfer relatively late in the process. I will tell you that when we layed in the initial B-1B program, I consciously deferred that planning because I didn't have the insight to plan

for it earlier. When we defer the planning, we also typically don't plan for the training for those maintenance resources. Frequently, by the time we lay in our plans to train software, the software maintenance people, the bright young men who developed the software are long gone and some of that intellectual knowledge cannot be recaptured for the training process. We also typically ignore the requirements for that software maintenance support environment. The standard answer for a support environment, we call it an Avionics Integrated Support Facility, is to use residual assets from the development program, and I think most of you do that. Yet those are resources that were tailored to the development process and they are not particularly efficient for the routine maintenance process. Then I turn around and criticize the acquiring community. But, I tell you that I have not seen, until very recently, my own command stand up and establish the software support requirements for any system early in the acquisition phase. We just did a document for the small ICBM missile which included a software support concept. But, as far as I know, that's the only time my own command has bothered to define our requirements up front. I can tell you on the B-1 that we did not define our requirements until very late, after we had production contracts.

Now, besides the transition, a much more serious problem is the real resource constraints that we face in the DOD. We have been living fat for the last five years; we've got lean times ahead. The Air Force Logistics Command has been reduced in manpower every year for the last two years, and I am confident that the Command will continue to shrink in manpower over the next five years. Now, I mentioned that we have about 3200 equivalent slots either in organic resources or Operations and Maintenance funding. I'll tell you also that number is an average of an 11% per year growth in manyears devoted to computer resource support for the last five years. We anticipate the computer resource support will continue to grow at the rate of about 11% a year. But that's not the growth in requirements. To give you an example of growth of requirements, in FY86 our requirements for Operations and Maintenance funding to support computer resources was $300M; of that, we only funded the portion that we deemed absolutely critical: $80M. The other $220M worth of unfunded software support were, in fact, many of the things that probably our customers deem as important things to have, not absolutely critical. Also, it's the development of the tools and the things that make our software support more productive - our own requirements that we deferred for those absolutely critical user requirements.

We're not going to get more money; we're not going to get more people; the requirements are growing. The answer that we all recognize is that the resources to do computer resource support, computer systems support, software support, are coming out of other core logistics functions in the Command. We get laughed at for $7,000 coffee pots and $700 hammers and the reason for that is the wrong level of manning in our inventory management area, the wrong skill level and grade structure in inventory management, and archaic computer systems to support those people. But it's the guys who are buying the $700 hammers and $7,000 coffee pots who are the people who we are taking resources away from, and infrastructure away from, to spend on computer resource support. That would be an easier tradeoff to make, and for management to swallow, if it were not for a real perception of waste in our software support

environments. I mentioned that, typically, our software support equipment facilities are residual development assets. As a result, they are very complex, expensive to maintain, they have rather narrow functionality, and it's difficult for one of our software engineers to stay proficient across a number of environments, so we get relatively low utilization out of both people and the equipment, except on some systems. Those facilities have an extremely high overhead. We just finished a new operational flight program for the F-15 at Warner-Robbins, and we devoted as many man-hours in updating the support environment to the new configuration as we did actually changing the operational flight program itself. That's not unusual, and I would submit, Tom (Smith, NAVAIR), that you probably see something in the same ratio in many of your facilities. Our software support facilities are not really tailored for productivity. Our tools are relatively clumsy in many cases.

The other thing that eats up a lot of our resources is that, although many programs are written in high order language, because of the growth and deferred requirements, once the operator gets the system and begins using it, we rapidly saturate the computer resources and we're driven immediately into assembly language. Typically, we can anticipate from the time that we get a system, within five years we will be driven into writing that entire OFP in assembly language to keep it within the computer. The F-15, again, is going through a computer upgrade; we reckoned that the requirements already on the book will totally saturate that upgraded computer within about 18 months. (Question directed at someone in audience: How long is it taking you to saturate the computer you're putting on the F-111? ANSWER: The day we put it on.) I'm sure that's not a unique story.

The real problem is productivity. Now, all of you have heard the same solution to this problem that I have, you know: Ada. I get the Ada zealots talking to me frequently. I've watched the STARS program, which is supposed to develop the basic technology for productivity, become totally an Ada support activity. I've watched the Software Engineering Institute, who is supposed to transition that new technology to aid productivity, become essentially an Ada support activity. In my support environment, I have pretty good insight into what my workload's going to be for the next ten to fifteen years because it's all sitting out there in programs at Systems Command. About 150 new processors, that I'm not supporting now, are somewhere in that acquisition pipeline. And I know how effective Ada is, and I know the probability of me retrofitting high capacity processors on most of the Air Force aircraft to handle the Ada overhead and I'll bet anybody in this room a bottle of good Scotch that in the year 2000 not more than 20% of our software in embedded computer systems will be written in Ada. I will still be supporting the old languages, primarily FORTRAN, Pascal, and assembly language in any number of applications. So, the people who walk around promising me the Utopia of Ada, frankly, have no credibility. We're not working the problems that Air Force Logistics Command, as the supporter, and you, as a supporter, are going to live with for the next fifteen years.

The final issue, or issue of concern -- crisis -- from the supporter's perspective ties back in with that operator view, and that's responsiveness. We all sell, and we're hawking software and software-intensive systems; we talk about the ease with which we can change the systems. That's probably a

good selling point, but it may be like talking about quality on a 1980 vintage General Motors car. I'm not sure that we're being totally honest with ourselves. The only place in the Air Force Logistics Command we practice rapid software change is in the Electronic Counter Measures environment. We do have a process and we do exercise it as part of various exercises -- getting in the data, generating very rapidly a change to an emitter threat, and transmitting that to the operational units so they can reburn the PROMs in the Electronic Counter Measure pods, and we generally do pretty good against the scenario. I will also tell you that it's a very structured exercise and probably lacks a lot in realism. Going back to that F-111 penetrating at 100 feet; as far as I know, and I believe this to be true, the software people on the F-111 have never once practiced a rapid reaction combat driven change to their OFP. The same is true for practically every other embedded computer system in the Air Force. We're taking slots out of our hide in the headquarters, establishing intelligence organizations down at our Air Logistic Centers, to make the first step in getting that intelligence data down to the software area, so we have the basic information to generate the change. But there are a lot of resource issues in structuring ourselves to be able to respond rapidly to a combat change in software, and I'm not sure that we in the Air Force Logistics Command can stand up for those resource requirements to really have a rapid reaction software change capability. I'll also tell you that we don't design our systems for that capability. The AMRAAM Missile, which is not only going to be the Air Force's new medium range missile, but is going to be the Navy's new medium range missile, the Marine Corp's medium range missile on the F-18 and for most of NATO: that missile is software intensive, and has very complicated Electronic Counter Measures requirements built into its guidance algorithms as you'd expect. The computer resources in that missile are all burned hard into PROMs and a sealed guidance section. A software change on that missile requires five years and requires 10% of all the Air Force's assets to be routed back through depot for the PROMs to be changed out. That's not an acceptable answer if we go to war. We're working the problem; we're looking at a change that allows us to use electronically erasable PROMs, but that's new technology that was not really available when the design of the AMRAAM was carved up. I suspect all the other Services have equivalent examples.

The other issue on the responsiveness side is that, even when we change the software, that's just the tip of the sword. There are a lot of other resources that have to be deployed with that software change, like the tech orders. On the F-16C/D right now we're going through a process out of the SPO of very rapid changes in the computer resources on that aircraft. We've had to resort to what we call a "walking tech order." Only when we get the OFP changed and released can we begin rewriting the tech orders to show the maintenance guys what the new diagnostic processes are on the aircraft, for example, and we are probably five months from getting those tech orders published, distributed and out to the mechanic. So, every time they release an OFP, we send a contractor out to the field to each one of the maintenance units to be a "walking tech order," to be able to explain what the new system means and why the old diagnostic steps don't work with the new software release.

That seems like an appalling catalogue of problems, and it's true. It is. What do we do? If I was talking to another forum besides software practitioners, I'd have one list of recommendations. I'd tell them to learn more about software -- to learn to speak software language. There's a lot of work that needs to be done to bring computer literacy to people outside of our community. I recommend, for the senior managers, that we make a very conscious effort to grow a generation of managers -- some of them are sitting out here -- who understand the software issues, who grew up in this environment and who can match both the shortcomings and the capabilities of the software with the overall operational requirements. I'd recommend to the users that they work much harder in defining what their real requirements are in being able to articulate that flexibility they need in their software to be able to fight the war. More than anything, I'd recommend that we in the Air Force and the Air Force Logistics Command and DOD develop a strategic vision of where we should be going in this whole computer resources Post Deployment Software Support arena. The ideas are bubbling up from the bottom but there's no overall vision at the top to direct those ideas and separate the good from the bad. But I'm not talking to managers outside of the software community.

I'm talking to practitioners, and I've got a different set of recommendations. The first one, which I believe is the most important, is to concentrate on the tools. You know, if we look at how we produce software, it's very analogous to the way the British army used to produce the old Brown Bess musket of Revolutionary War vintage. Each one of those was hand-built, and if you could take a piece off of one and fit it on another one, you were damned lucky. In fact, typically, if you broke a hammer or you broke a spring or a screw, you gave it back to an armorer who went out and literally built you another piece for that particular gun. What we need to do in the software arena -- and this is not a new idea on Lew Curtis' part -- is to do the same thing Eli Whitney did for the manufacture of muskets, or rifles, when he developed basically the production line and the use of interchangeable parts and made the musket a much more supportable weapon out there for the soldiers. We need to make software development and maintenance a science and not a craft, and that's really what it is now, a craft. We're back to the issue on productivity.

The second recommendation I would have for the practitioner is to make a conscious effort to speak the language of management. I've been at briefings where very, very important issues were being addressed, but because the individual briefer was speaking in his own software vernacular, the comprehension rate was probably not ten percent. Some of you have probably seen the exact same thing. Many of us, myself included, tend to think of software maintenance in terms of hardware analogies and it leads us astray. The software world needs to be able to communicate better with the non-software world.

Another foot stomper is: being sensitive to your customers. Colonel Reed and I were talking about that earlier. Understand what your customer's real requirements are. Help him define his requirements. Interact with your customer. He doesn't know what your capabilities are, and until he understands the real capabilities, he can't really articulate what his requirements are.

Work todays problems. Ada, STARS, and the Software Engineering Institute are all great initiatives, but we also need to temper that vision of the perfect data world with the real world that exists today, and work today's problems. The bottom line all of us have to remember, and I'd say across every one of our systems, is that the basic purpose of the system is to fight a war. We must remember the requirements of obtaining a war fighting capability, or enhancing a war fighting capability, of our system. I'm very encouraged by, first, the existence of this Orlando II conference. I think it's an absolutely essential step. I'm tickled to death with the conversations and the progress I've seen in the panels. I think the fact that we can recognize the issues we're working and address the recommendations, taskings, and plans is a very critical first step in achieving some of the things I've talked about. But, all of you probably realize that the charter of this activity is really daunting. It reminds me of a story. God came down one day, as He does occasionally, and decided to walk a city street and see how us poor human beings were getting along. And, as He walked along, there was a little boy sitting on the curb, and the little boy had his head in his hands and tears were running down his face. And the Good Lord asked the boy what his problems were, and the little boy gave Him this long litany of the problems that a six year old would have. The Good Lord patted him on the back and said, "Don't worry, my son, all your problems are solved." The little boy brightened up and smiled, and proceeded on his way. A little later, further down that street, the Good Lord ran across a ten year old little girl with a long face, pouting, obviously in distress. He asked her what the problem was. Same thing. She explained her problems. He said, "Ah, my child, do not worry. Peace be with you. All your problems are solved." She smiled and proceeded on, perfectly happy. A little further down that street, the Good Lord ran across a software engineer. Same scenario. The software engineer explained his problems to the Good Lord, and the Good Lord sat on the curb and cried. We can share our problems, but only we can solve them. Thank you very much.

(INTENTIONALLY BLANK)

BRIGADIER GENERAL ALONZO E. SHORT, JR.

Brigadier General Short is currently the Deputy Commanding General/Deputy Program Manager of Army Information Systems for the United States Army Information Systems Engineering Command, (ISEC), located at Fort Belvoir, Virginia.

Brigadier General Short was born in Greenville, North Carolina, on 27 January 1939. Upon completion of the Reserve Officers Training Corps curriculum and the educational course of study at Virginia State College in 1962, he was commissioned a Second Lieutenant and awarded a Bachelor of Science degree in Industrial Arts. He also holds a Master of Arts degree in Business Administration from New York Institute of Technology. His military education includes completion of the Signal School, the Armed Forces Staff College, and the United States Army War College.

He has held a wide variety of important command and staff positions culminating in his current assignment. Immediately prior, he served as Commander, United States Army Information Systems Management Agency/United States Army Electronics Systems Engineering Installation Activity/Project Manager, Defense Communications Systems, Fort Monmouth, New Jersey. Other key assignments held recently:

Deputy Commander, USA Electronics Research and Development Command (ERADCOM), Adelphi, Maryland.

Commander, 3d Signal Brigade, Fort Hood, Texas.

Assistant Corps Communications and Electronics Officer, III Corps and Fort Hood, Fort Hood, Texas.

Chief, Plans Branch, Operations Division, later Chief, Plans and Requirements Branch, Plans Division, Office of the Deputy Chief of Staff for Operations and Plans, United States Army Communications Command, Fort Huachuca, Arizona.

Student, United States Army War College, Carlisle Barracks, Pennsylvania.

Special Assistant to the Chief of Staff for Reforger Operations Planning, 101st Airborne Division (Air Assault), Fort Campbell, Kentucky.

Commander, 501st Signal Battalion, 101st Airborne Division (Air Assault), Fort Campbell, Kentucky.

Deputy Program Manager, Secure Voice Division, Defense Communications Agency, Washington, D.C.

Awards and decorations which General Short has received include the Legion of Merit and the Bronze Star Medal (with Oak Leaf Cluster). He is also authorized to wear the Parachutist Badge and the Air Assault Badge. Other awards include the National Defense Service Medal and the Republic of Vietnam Ground Combat Medal.

General Short and his wife Rosalin (Roz) have two children: Stanley and Daniele.

SPEECH BY BRIGADIER GENERAL ALONZO E. SHORT, JR., USA
Deputy Commander, Army Information Systems Engineering Command
Wednesday, 28 January 1987

I am delighted to be here in Orlando today to address this luncheon session of the Joint Logistics Commanders 4th Biennial Software Workshop.

From your workshop brochure and schedule, I see that your purpose and objectives are lofty and challenging, but I also note with confidence that, observing the list of highly qualified participants, these challenges will be met and exceeded.

When I was asked to substitute for Major General Alan Salisbury as the Luncheon Speaker at a software workshop, two emotions immediately gripped me--elation and trepidation. If you didn't already know--Gen Salisbury is perhaps the Army's leading authority on software and software development. Unfortunately he is still recuperating from pneumonia or he would be here today. As for me, after a week of snow in the National Capitol Region the warmth, hospitality and support down here have been very comforting.

It is indeed gratifying for me to tell you today that the Army recognized the magnitude of the software problems, particularly in post deployment support and has committed to a strategy to meet these challenges.

In May 1984, the Chief of Staff Army (CSA) made the decision to reorganize the Army and establish the Information Mission Area (IMA). Establishing the ACSIM--at the DA staff level--for three star level policy making and the Information Systems Command (ISC) for three star command implementation throughout the Army. Gen Salisbury's command (ISEC) is the principal organization in ISC charged with developing standards, and performing hardware and software engineering and integration to meet the challenge. I'm happy to report that most of our planning is complete and execution has begun.

Concerted efforts by the Army leadership have caused an Army wide commitment to the IMA strategy. This strategy calls for the integration of the five IMA disciplines; namely, telecommunications, ADP, records management, audio/visual information and publications and printing and the three environments; namely, strategic, theater/tactical and the sustaining base into a coordinated and integrated Army Information System (AIS). In this regard, one of ISEC's most important missions is to develop and recommend appropriate standards across the disciplines and environments. Our PM's, engineers, SDC's, and field commanders are all involved in the process.

Standards are vital to the development of the Army Information Architecture. It has been through standards that an abstract conceptual three tiered architecture has become an implementation reality. The three tier architecture is relatively arbitrary, but is a useful baseline and reference point to achieve common understanding.

The Army recently announced a comprehensive set of standards for its information systems. I'd like to point out, however, that the standards selected are also industry/commercial standards, namely,

o	Tier 1 (RDC Large Mainframes)	MVS,
o	Tier 2 (Installations/Agencies)	MVS and UNIX,
o	Tier 3 (Individual Work Stations)	MS DOS and UNIX for DBMS realtional with SQL and the communications standards (SNA evolving to OSI).

Under this philosophy, we ensure competition without the continuing necessity of costly and consuming conversion and/or interface development. Moreover, the standards provide the Army with the maximum freedom from dependence on any one vendor for any one product or class of products.

The general approach is data centered, not process centered. We see data as a common resource which must be used and managed by all. In ISEC we have begun modernizing the STAMIS by separating data from application and enforcing data standards. We are slowly but surely eliminating costly unique or "stovepipe" systems. A bold move toward modern data systems.

Our thrust emphasizes end user programming in order to control resource costs, produce smaller and easier-to-write programs, and to converge requirements (that is, capabilities that function both in the office and on the battlefield).

The idea and move to develop an Army Corporate Data Base (ACDB) is also an integrator because all users and developers can use the same data, even though the view may be different. We are "priming the pump," and the need is definitely for smart tools that both help the end user and capture and develop experts.

Acquiring smart support tools and products is an Army priority. Training, however, is everyone's concern. To this end we are agressively pursuing our transition to Ada, for it's no longer just the wave of the future, Ada is the standard for the Army. Exceptions or waivers will not be granted or one must go through an arduous process to obtain one. We are actively training our developers, acquiring tools, and contractng to transition existing systems to Ada.

Many ask, why Ada as a standard?

- o Only language designed to meet DOD requirements
- o Offers significant technical management advantage
- o Definition controlled by DOD
- o Compiler compliance controlled by DOD
- o Modern software engineering encouraged
- o Promotes software reuse

With a significant number of certified Ada compilers in use now, we will push on with Ada for the Army's STAMIS and are prepared to accept unavoidable schedule delays associated with learning curve of new programming language. The same can be said for the use of Structured Query Language with DBMS.

Standards. I'll conclude by saying that the Army understands the need for fully integrated solutions that can meet its information processing and transfer requirements. We are committed to the standards and are moving with vigor to implement them. The challenges to ISEC as a professional organization are numerous, as are the benefits that will accrue to the Army. Again, thanks for this opportunity to share a few thoughts with you on where the Army is going with information systems standardization.

(Intentionally Blank)

REAR ADMIRAL HARRY S. QUAST

Rear Admiral Quast is presently assigned to the Office of the Chief of Naval Operations as Director, Information Systems Division (OP-945). He also serves as Director, Department of the Navy Information Resources Management under the cognizance of the Assistant Secretary of the Navy (Financial Management).

Rear Admiral Harry S. Quast is a 1957 graduate of Miami University in Oxford, Ohio where he participated in the regular Naval Reserve Officers Training Course (NROTC). Upon graduation, he was commissioned an Ensign and reported for duty aboard the USS SAN MARCOS (LSD-25), assigned as Navigator.

From 1960 to 1961, he served as an NROTC instructor at the University of Louisville in Louisville, Kentucky. From there, he reported as Operations Officer, and later Engineering Officer, aboard the USS HAZELWOOD (DD-531). In June 1964, he attended the U.S. Naval Postgraduate School, Monterey, California, where he earned a Master of Science degree in Business Data Processing. From Monterey, then Lieutenant Commander Quast reported aboard USS SKILL (MSO-471) in Charleston, South Carolina, as Commanding Officer.

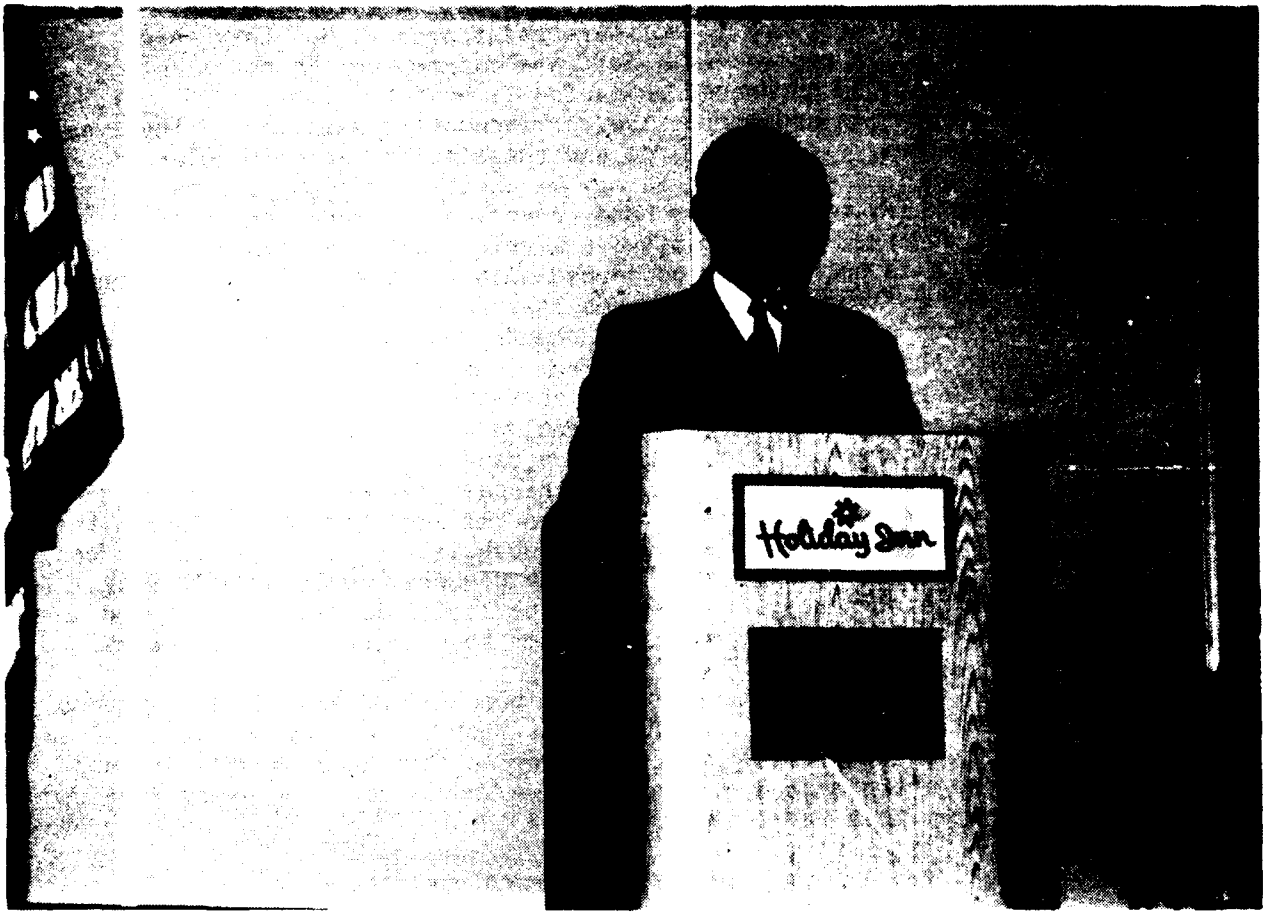
Following a two-year tour at the Naval Military Personnel Command, he assumed command of the USS HULL (DD-945) in San Diego, California. Following that assignment, he was assigned to the Industrial College of the Armed Forces at Fort McNair in Washington D.C.

After another assignment in the Naval Military Personnel Command, he was assigned as Assistant Chief of Staff, Manpower and Personnel, to the Commander-in-Chief, U.S. Pacific Fleet, from 1978 to 1980. His most recent sea assignment, serving as Commander, Destroyer Squadron FIVE, home ported in San Diego, California, from July 1980 to July 1982, followed.

Rear Admiral Quast holds both the Meritorious Service Medal and the Bronze Star with Combat "V". He is a native of Sheboygan, Wisconsin, and is married to the former Merril Sleight of San Diego, California. They have three children: Harry, Catherine, and Jennifer.

(Intentionally Blank)

SPEECH BY REAR ADMIRAL HARRY S. QUAST, USN
Director, Information Systems Division
Wednesday, 28 January 1987



Why are we here at this workshop? Let me tell you why I think we are here. We are here because software is critical to every system. We are here because we presume that we can develop and maintain software better than we are currently doing. We are here because we think we can improve the state of the practice. According to the 1985 EIA study, DOD will spend approximately 20 billion dollars on MCCR in 1987. This number will climb to over 42 billion dollars by 1995. The EIA report also noted that the percentage of the MCCR cost attributable to software is approximately 85% of the total MCCR costs. What is more dramatic is that the EIA report shows that hardware costs will rise approximately 78% between 1985 to 1990 and only 21% between 1990 and 1995. This indicates a leveling of hardware costs. But what do you think they said about software? You're right, software costs are going to go up by 100% in the period 1985-1990 and will then go up another 100% from 1990 to 1995. For software, there does not seem to be any leveling. This is serious! Software is the critical element that we have to be concerned with, as it is going to eat up a larger and larger percentage of our budget, and this is going to make it very difficult for us to meet all of our mission requirements in the future.

Software is the brains and nervous system of our systems, yet we have to afford them to have them. Post deployment support, as you are aware, has been estimated by different people to consume from between 50 to over 70 percent of the software life cycle cost. If this is true, and I have no reason to doubt it, then we have to address and solve this problem as soon as possible.

Software seems to be in the position of the tail wagging the dog. The most expensive item in software is software maintenance. But this term, software maintenance, is misleading. Software maintenance is not like hardware maintenance. In hardware, we basically have two types of maintenance, preventative and corrective. Preventative maintenance involves replacement of filters, cleaning boards and chassis, and doing regular diagnostic tests to isolate parts that may be on the verge of failure. Corrective maintenance, on the other hand, involves the replacement of a failed component or assembly. In software I think that what comes under this term of maintenance is not as much concerned with failure as it is concerned with change in the software that increases its usefulness and its functionality. When mission requirements change, we change the software and we lump this into the term maintenance. From some of the studies that we have looked at, only 20 percent of what we call software maintenance is related to what could be called preventative and corrective maintenance. The other 80 percent is new development. This can be very dangerous, as software maintenance has not traditionally followed the same degree of discipline as we have applied to software development. We have all come to the conclusion that for software development to have as high a probability of success that it can, management discipline is vital. This is a key cornerstone of DOD-STD-2167. Yet with software maintenance, it seems that we lack this discipline. And software maintenance is not that much different than software development.

I commend you for starting to attack this problem. I also implore you to help us get a better handle for this difficult task as soon as possible. I am afraid that we can't allow PDSS to grow any larger than it currently is, and yet all of the projections that have come to me show a doubling every five years. Help us stem this unbridled growth.

I want to tell you a short story that I heard over the past few weeks and it will illustrate an important lesson that we have to consider during the rest of this workshop.

"It is the fourth quarter of the football game, and Podunk University is ahead by 6 points. The ball is on the team's 20 yard line and the first-string quarterback gets injured. The second-string man is also hurt, so the coach is forced to use his third-string quarterback. The coach pulls the young man aside and says, "Now listen kid, go in there and do two quarterback sneaks and then punt, no matter what! I hope you understand, two quarterback sneaks and then punt, no questions, just do it!"

The quarterback nervously takes the snap, finds a huge hole, and runs for 40 yards. On the next down, he takes the ball and runs with it, breaking loose for 37 more yards. Now, on the opposing team's three yard line, the quarterback drops back and punts.

The coach jumps off the bench and screams at the quarterback now coming off the field. "You idiot, what were you thinking?" With a deep sigh the quarterback says, "I was thinking: 'What a dumb coach!'"

Let's not make the mistake of this coach and give people no options. As I previously pointed out, post deployment support is a costly item and the cost seems to have no end in sight. We want a solution so badly, yet, I am afraid that we may strive for a cookbook solution when no cookbook solution is good for all the PDSS problems. You probably have looked at many different environments and have noticed many similarities and differences. Your groups have been designed to address PDSS problems throughout the life cycle. One group is devoted to PDSS Planning During Development while another is concerned with Forecasting PDSS Resource Requirements and yet another is concerned with the issues related to Human Resources in PDSS. You also have groups addressing the important problems associated with the Software Change Process and a group looking at PDSS Standards. I also noticed that there are panels on PDSS Management Indicators and Quality Metrics, Software Technology Transition, and Mission Critical Computer Resource Security. Each one of these groups is critical. We need solutions.

Please let me become a little parochial and discuss some of our thinking within the Navy concerning the overall MCCR environment. One key element of our thinking is the need to support our systems no matter where they may end up. Some will be ashore while others will be on surface ships and others undersea. Some of our systems are aboard aircraft while others are in space. The logistics headaches that this unique situation produces has caused the Navy in the past to look at hardware standardization as a partial solution. But we now have some elements of the environment that are hopefully going to not only improve our software development, but also aid us in developing more maintainable software. I, like my counterparts in the other Services, feel that Ada is a plus. We, in the Navy, have committed ourselves to the use of Ada in all of our weapon system developments. We are developing the only Service developed and supported Ada support environments, the ALS/N. The effort is proceeding and I hope that standardization in this case will also be a plus. It should be noted that the ALS/N, like any system, will need to be maintained. The ALS/N, like the application systems that you are investigating, will need a PDSS. I do not envision the ALS/N to be stagnant. It will have to develop and grow as we learn more about PDSS needs and as technology progressess. As is evident, software maintenance of the ALS/N involves continued development along with preventative and corrective maintenance actions. In that regard, support software doesn't appear any different than application software.

Our commitment to the ALS/N fits in very closely to our commitment to the DOD software initiative programs. As I said, we have committed ourselves to Ada. We are also strongly committed to making the DOD Software Engineering Institute a success. I am a member of the SEI's JAC-EG and, in the interactions that I have had with the SEI, I am hopeful that the programs at the SEI will benefit us all. The SEI has an ambitious program and it has the support of each of the Services. This now brings us to the last element of the DOD software initiatives, STARS. STARS has travelled a rocky road to date and it still isn't totally on course. But STARS is in my opinion a very critical element of our overall software strategy. STARS is planned to help us develop new products, concepts, and approaches. The SEI is designed to

help integrate these into defense systems. And the cornerstone of the near and mid term solutions is Ada. To increase our probability of success, we have to all work as one team, with one objective, and get these working together. I hope that this will occur.

The JLC Joint Policy Coordinating Group on Computer Resource Management has a tradition of successes. Back in the 1970's, we had standards and DIDs coming out of our ears, everyone had their own. The CRM was established to address and solve this, as well as other problems. You first attacked the standards issues, and as you all know, DOD-STD-2167 is a testament to the CRM's successful accomplishment there. I understand that a revision of 2167 is on the verge of being released. This revised 2167 will more strongly address software quality management. I think that this is important and I support your efforts. 2167 and its revision are the by-products of workshops like this. I am confident that, as in the past when we put our best talent to work on these problems, we will overcome adversity and succeed in moving forward and improving the state of practice in software.

I want to wish you a productive meeting. Thank God we have people like you, dedicated, understanding, and knowledgeable. I am confident that as in the past, the problems confronting us with respect to the PDSS problems will be solved in the near future.

MAJOR GENERAL MONROE T. SMITH

Major General Monroe T. Smith is Deputy Chief of Staff, Product Assurance and Acquisition Logistics, Headquarters Air Force Systems Command, Andrews Air Force Base, Maryland.

General Smith was born July 17, 1931, in Glenwood, Georgia, and grew up in Plant City, Florida, graduating from Turkey Creek High School in 1948. He enlisted in the U.S. Air Force in February 1951 and served as a Flight Line Crew Chief from 1951 until 1957. He was serving in the grade of Technical Sergeant when he entered Officer Candidate School, graduating in March 1958.

From April 1958 to December 1958, General Smith attended Maintenance Officer School at Chanute Air Force Base, Illinois. In January 1959 he joined the 4130th Strategic Wing at Bergstrom Air Force Base, Texas, as a Flight Line Maintenance Officer. He completed Squadron Officer School by correspondence in 1961. From September 1961 to July 1964, the general served with the 577th Strategic Missile Squadron, Altus Air Force Base, Oklahoma. He moved from Deputy Commander of an Atlas F missile launch crew to Crew Commander, and for his last year, served as the Senior Instructor Crew Commander. He was then assigned as a Maintenance Staff Officer for Policy and Procedures, in the Office of the Deputy Chief of Staff for Logistics, Headquarters Strategic Air Command, Offutt Air Force Base, Nebraska.

General Smith attended several colleges, graduating from the University of Omaha in 1966 with a Bachelor of General Education degree. He later graduated from The George Washington University, Washington D.C., in 1968, with a Master of Science degree in Public Administration. He completed Air Command and Staff College, in residence, at Maxwell Air Force Base, Alabama, in 1968. Following graduation from Air Command and Staff College, General Smith served with the 483rd Tactical Airlift Wing, Cam Ranh Bay Air Base, Republic of Vietnam, as the Wing Maintenance Control Officer until July 1969.

The general served as a Faculty Instructor at the Air Command and Staff College, Maxwell Air Force Base, Alabama, from August 1969 to August 1972. He then attended, and graduated from, the Air War College, also at Maxwell Air Force Base, in May 1973. Following graduation, he was named Commander, 22nd Organizational Maintenance Squadron, March Air Force Base, California. The general was assigned as a Maintenance Staff Officer in the Aircraft Systems Division, Office of the Deputy Chief of Staff for Systems and Logistics, Headquarters U.S. Air Force, Washington D.C., in September 1974.

In January 1975, he became Chief of Executive Services for the Deputy Chief of Staff for Systems and Logistics. General Smith

was selected as a research associate for the Fletcher School of Law and Diplomacy, Tufts University, Medford, Massachusetts, from July 1975 to August 1976. He then became Director for Plans and Industrial Resources, Headquarters Air Force Logistics Command, Wright-Patterson Air Force Base, Ohio.

Transferring to McClellan Air Force Base, California, in February 1978, the general served as Director of Materiel Management for the Sacramento Air Logistics Center. In March, 1981, he moved to Los Angeles as Commander of the Defense Contract Administration Services Region. He was then assigned as Deputy Chief of Staff for Maintenance at Air Force Logistics Command Headquarters in July 1982. General Smith completed the Advanced Management Course at Carnegie-Mellon University in Pittsburgh in 1980 and attended the Harvard Executive Program on National and International Security in 1983. In July 1983, he became Commander of the Air Force Acquisition Logistics Center, Wright-Patterson Air Force Base. He assumed his present position in July 1985.

His military decorations and awards include the Defense Superior Service Medal, Legion of Merit with one oak leaf cluster, Bronze Star Medal with one oak leaf cluster, Meritorious Service Medal with two oak leaf clusters, Air Force Outstanding Unit Award Ribbon, Good Conduct Medal with bronze clasp and three loops, National Defense Service Medal with service star, Vietnam Service Medal with four service stars, Air Force Longevity Service Award Ribbon with seven oak clusters, Small Arms Expert Marksmanship Ribbon, Republic of Vietnam Gallantry Cross with palm and Republic of Vietnam Campaign Medal.

He was promoted to Major General October 1, 1983, with date of rank September 1, 1980.

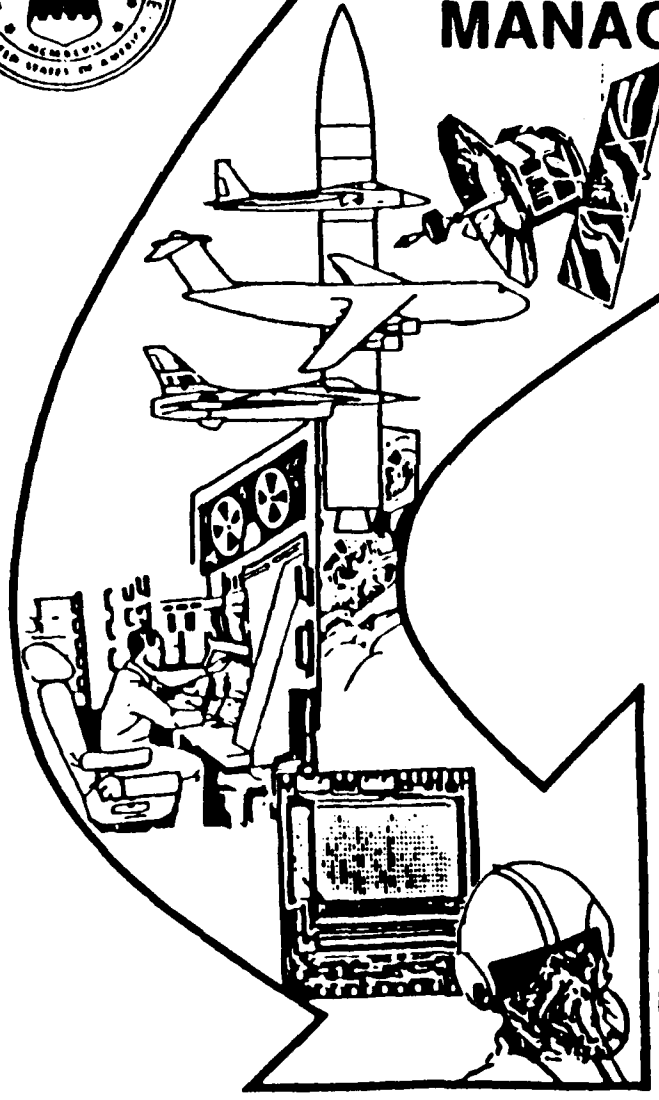
General Smith is married to the former Flo K. Parrish, also from Plant City, Florida. They have two children: Terry and Michael, both of San Francisco, California.

SPEECH BY MAJOR GENERAL MONROE T. SMITH, AFSC
Deputy Chief of Staff, Product Assurance and Acquisition Logistics
Thursday, 29 January 1987

Major General Smith presented the Air Force's posture concerning Software Management, utilizing both prepared text and viewgraph slides. Copies of those slides are provided on the following pages. Additional comments are provided on the facing page of those slides that Major General Smith discussed in detail.

(Intentionally Blank)

AIR FORCE SOFTWARE MANAGEMENT



BRIEFING

MAJ GEN M.T. SMITH
HQ AFSC/PL

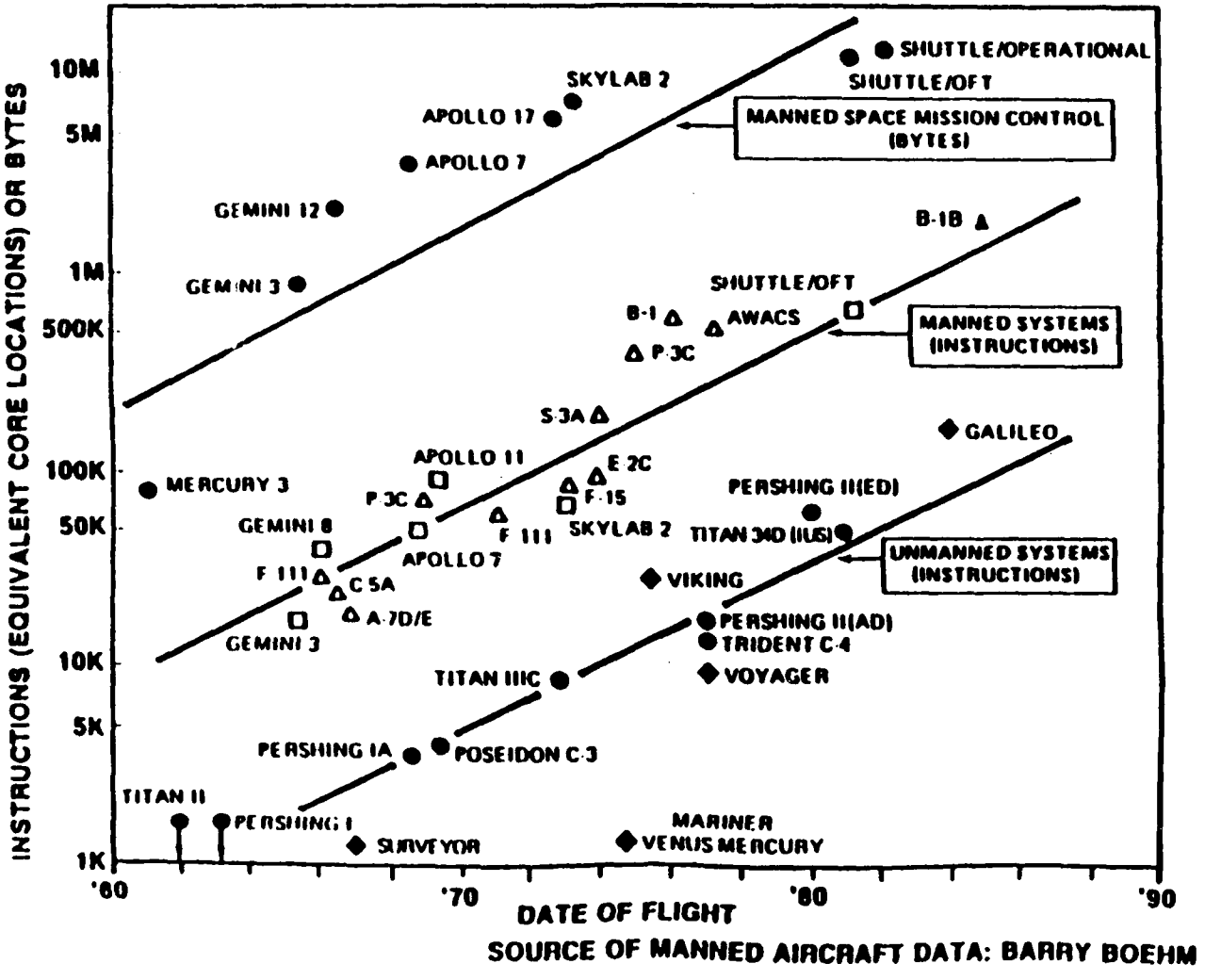
OVERVIEW

- BACKGROUND
- AIR FORCE INITIATIVE
- DOD SOFTWARE INITIATIVE
- SUMMARY

SOFTWARE MANAGEMENT PROBLEMS
THE SOFTWARE FORCE MULTIPLIER IS IN TROUBLE

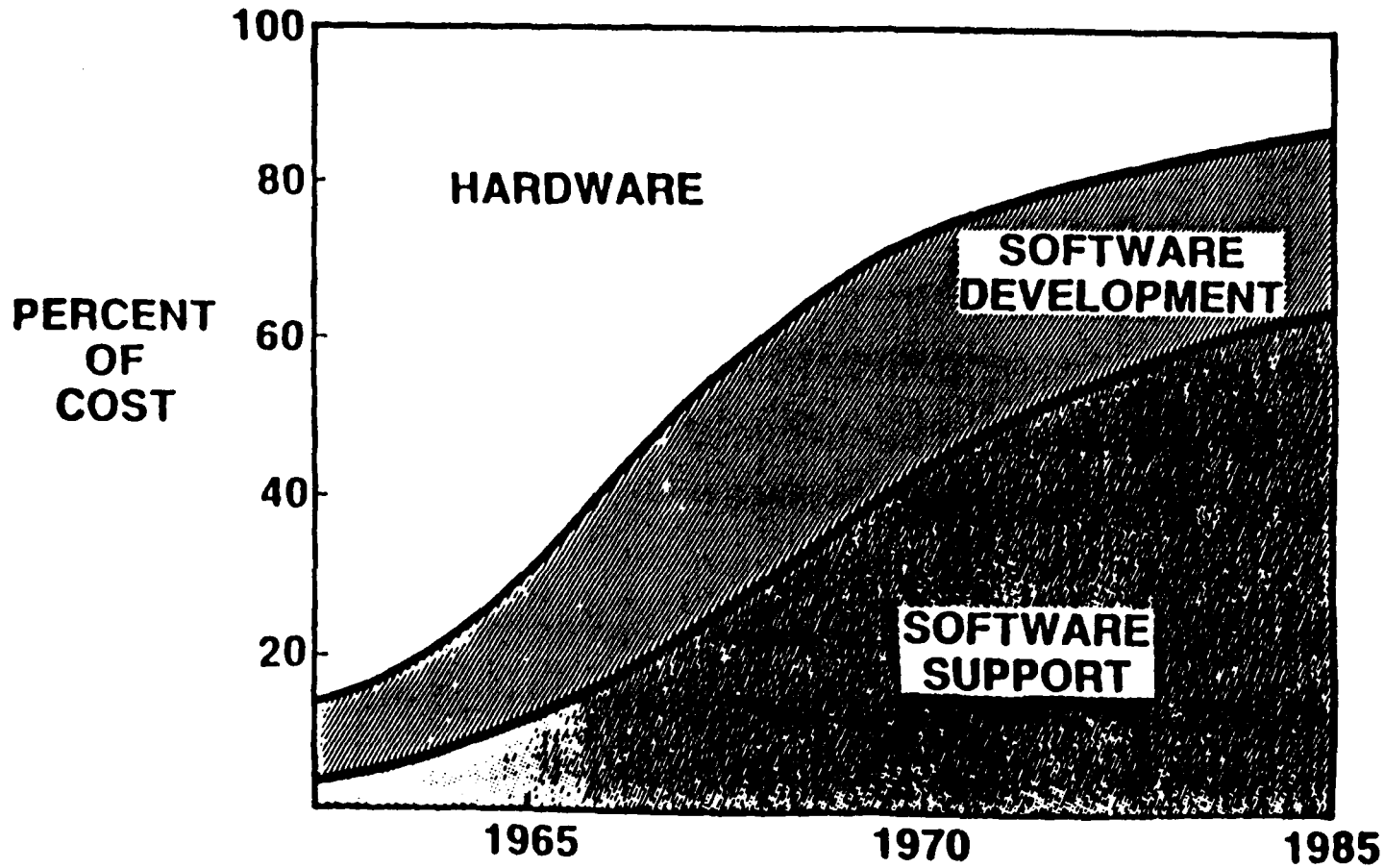
- Increasing demand for software
- Growth in software complexity
- Escalating costs
- Shortage of software professionals
- High cost of software support
- National problem
- No simple solution
- Bottom line: Software is the problem in seven out of ten troubled systems

TRENDS IN SOFTWARE



SOURCE OF MANNED AIRCRAFT DATA: BARRY BOEHM

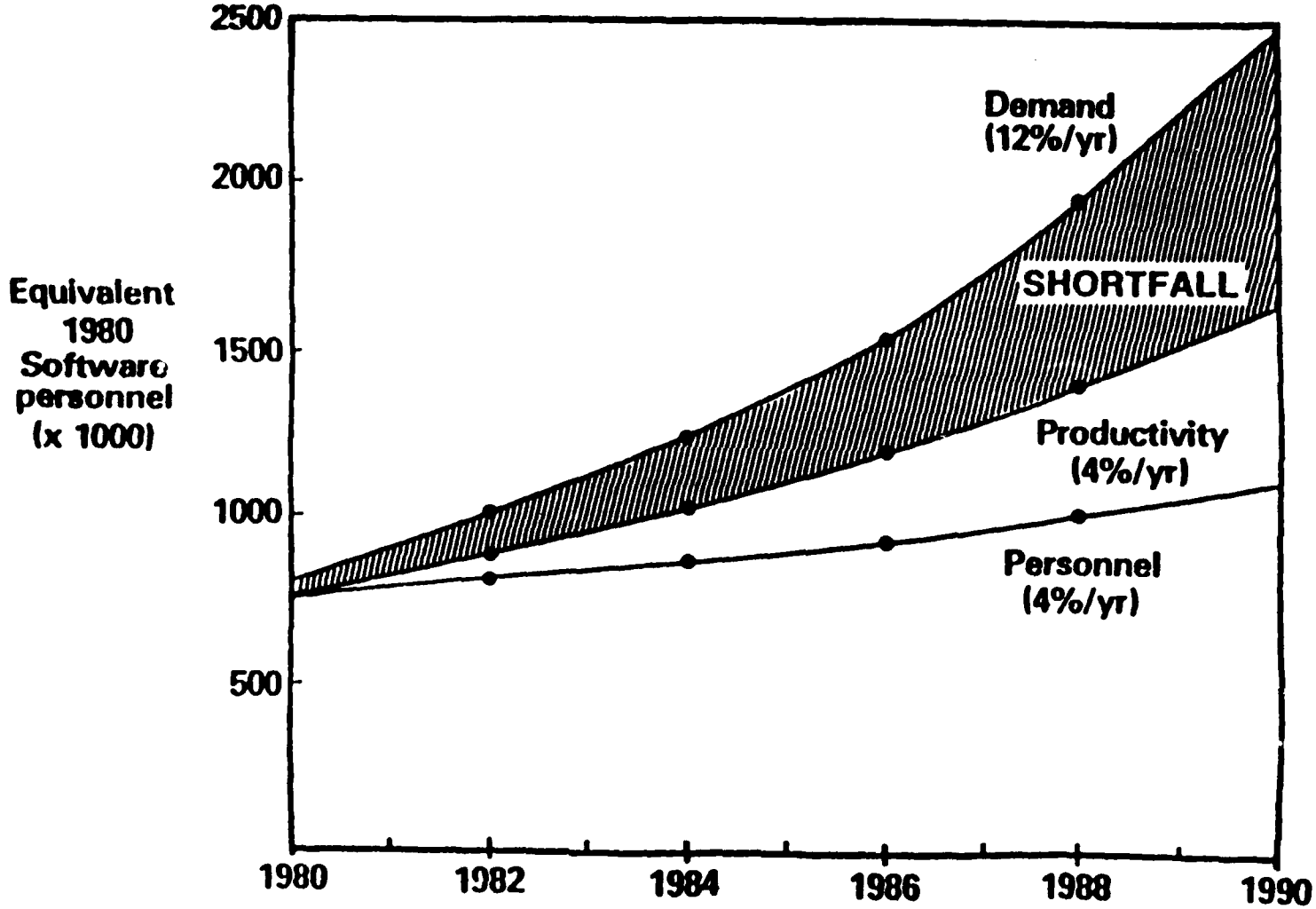
SOFTWARE SUPPORT COSTS ARE TOO HIGH



THE SOFTWARE CRISIS IS A NATIONAL PROBLEM

- DOD will be competing with its own contractors for software resources
- Foreign competition/intrusion could erode or interfere with advanced DOD software technologies

NATIONAL TRENDS IN SOFTWARE PERSONNEL SUPPLY AND DEMAND



SOFTWARE PERSONNEL SHORTAGE

- National Science Foundation 1987 predictions
  - 115,000 to 140,000 shortage of systems analysts and programmers
  - Will take 25-30% increase in supply to meet demand
- Electronics Industries Association predictions
  - 1,000,000 software professionals short to produce $32 billion of software for mission critical systems in the year 1990
- Air Force Systems Command
  - 94% manning for software in 1985
    - 68% of these are Lieutenants

- AIR FORCE INITIATIVES

SOFTWARE INITIATIVES -- DUAL SCREEN

The Defense System Software Development Standard, DOD-STD-2167, was the first major step taken to place discipline and visibility into software development. It describes a multiservice agreed to process for the development of quality and supportable software products. Since software development is a dynamic process, this standard is being revised to ensure that it accommodates new development techniques and to make it even more user friendly. This long needed tool improves the requirements specification and analysis process, reducing the quantity, while at the same time improving the usefulness, of software documentation, and establishing a structure for the associated data collection that will enable all of us to evaluate both software products and the software development process. Achieving supportable software depends upon the interrelationship of the process used to plan, build in, and maintain the quality of the products and evalution of the software products themselves. Our Software Management Indicators Pamphlet 800-43, addresses the process attributes. The indicators in this pamphelt build upon the structure provided by DOD-STD-2167 to provide management with tools to gain insight into the software development process. AFSCP 800-43 provides the field with a set of tools that they can use to manage software development. Now, we expect our program offices to manage software development proactively and not reactively.

While the AFSCP 800-43 indicators do address the process, visibility into the quality of the software products provided by these indicators is indirect. To address this short fall, we have recently published our software quality indicators pamphlet, AFSCP 800-14. This pamphlet follows the same approach we used with our original set of process indictors by looking at trend data, not software quality metrics. While the quality indicators are more technically involved than the management indicators, the intent is the same -- to provide the program office with another tool that will help in obtaining reliable and supportable software products.

Both AFR 800-14, Life Cycle Management of Computer Resources, and DOD-STD-2167 require the acquisition community to address software risk management. Unfortunately there is very little that describes what software risk management encompasses. AFSCP 800-XX is our approach to incorporating software risk management into overall system risk management.

AFSCP 800-YY is another initiative we are taking which will standardize the best features of the software pre-award surveys and capability/capacity reviews that are currently being done by our various product divisions.

HQ AIR FORCE SYSTEMS COMMAND SOFTWARE INITIATIVES

- DEFENSE SYSTEMS SOFTWARE DEVELOPMENT STANDARD (DOD-STD-2167)
- SOFTWARE MANAGEMENT INDICATORS (AFSCP 800-43)
- SOFTWARE QUALITY INDICATORS (AFSCP 800-14)
- SOFTWARE RISK MANAGEMENT (AFSCP 800-XX)
- SOFTWARE PRE-AWARD SURVEYS (AFSCP 800-YY)

INITIATIVE STATUS-DUAL SCREEN WITH THE HQ AFSC

(Text is Self Explanatory.)

An Ambitious, but achievable schedule.

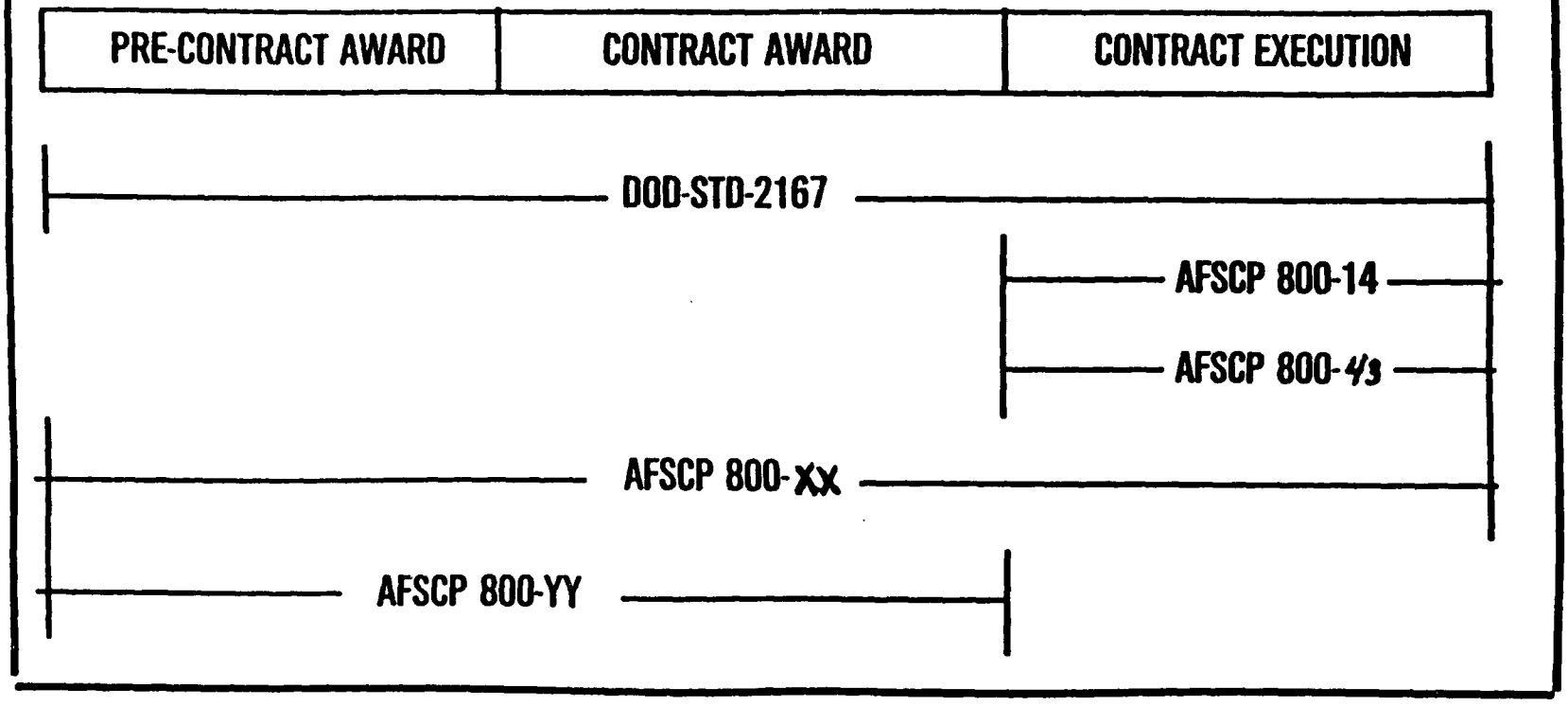
SOFTWARE INITIATIVES STATUS

INITIATIVE	STATUS
DOD-STD-2167	UNDER REVISION
AFSCP 800-43	PUBLISHED 31 JAN 86
AFSCP 800-14	PUBLISHED 20 JAN 87
AFSCP 800-XX	PLANNED JUN 87 PUBLICATION
AFSCP 800-YY	PLANNED MAY 87 PUBLICATION

INITIATIVES APPLICATION PHASES

Both DOD-STD-2167 and the risk management pamphlet will be applicable across the entire spectrum of software acquisition. The indicator pamphlets concentrate on the contract execution phase, while AFSCP 800-YY will be used to determine whether or not the development contractor has the capability to meet our software requirements from process, product, quality, and supportability perspectives.

INITIATIVE APPLICATION PHASES



SOFTWARE MEASUREMENT & SOFTWARE QUALITY (DUAL SCREEN)

Software measurement is one of the keys to achieving supportable software products. In 1891 the famous British Physicist Lord Kelvin said, "When you can measure what you are speaking about, and express it in numbers, you know something about it. But when you cannot measure it, when you cannot express it in numbers, your knowledge is a meager and unsatisfactory kind. It may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stage of science." In order to be able to measure software we need management commitment to use the tools we have developed under our various initiatives as well as the ability to express in numbers the quality of our software processes and the resultant software products. Through the application of our initiatives we hope to take the first bold steps in moving software development and aquisition from the "Black Arts" to a "Quantifiable Science" that will produce affordable and supportable software.

SOFTWARE MEASUREMENT ENTAILS MANAGEMENT AND QUALITY

ACHIEVING SOFTWARE MEASUREMENT

MANAGEMENT

ESTABLISH THE FRAMEWORK FOR MEASUREMENT

ATTRIBUTES OF
- MANAGEMENT PLAN
  - CONTRACT REQUIREMENTS
  - MANAGEMENT COMMITMENT

ADDRESSED BY
- DOD-STD-2167
- AFSCP 800-14
- AFSCP 800-43
- AFSCP 800-XX
- AFSCP 800-YY

QUALITY

EXECUTE PLAN AND CONTRACT

ATTRIBUTES OF
- PROCESS
- PRODUCT

SOFTWARE QUALITY ENTAILS PRODUCT AND PROCESS

ACHIEVING SOFTWARE QUALITY

PRODUCT

PLAN AND PERFORM QUALITY EVALUATIONS

ATTRIBUTES OF

- DOCUMENTATION
- DESIGN
- CODE
- TEST

ADDRESSED BY

- DOD-STD-2167 (2168)
- AFSCP 800-14

PROCESS

PLAN, BUILD IN, MAINTAIN QUALITY

ATTRIBUTES OF

- PEOPLE
- METHODS
- ORGANIZATIONS
- TOOLS
- FACILITIES

ADDRESSED BY

- DOD-STD-2167
- AFSCP 800-43

DOD SOFTWARE INITIATIVES

Next, I want to describe three advanced development programs sponsored by the OSD. These three programs are complementary thrusts that the DOD is using to find and to feed high technology into the software development process.

- DOD SOFTWARE INITIATIVES

DOD SOFTWARE INITIATIVE

When we overhaul a mission capability that's largely based on software, we cannot afford the dollars, or the time, or the complexity of mostly manual rewrites, from the ground up, of all of the old code.

Let's see how the three programs of the DOD Software Initiative will enable us to make responsive, timely, cost-affordable, parts-based software upgrades to support improved mission capabilities.

The three programs are Ada, STARS, and the SEI. Let me make clear what these programs are, and what they do.

You know that Ada is a standard DOD language, but you should also know that the Ada Program is the keeper of MIL-STD-1815. It works towards assuring that Ada becomes the common base for developing and supporting mission critical software.

The STARS program, under its new leadership at OSD, has taken on an Ada focus for its work, which is inventing (and demonstrating) the automated tools and reusable parts which let us use Ada.

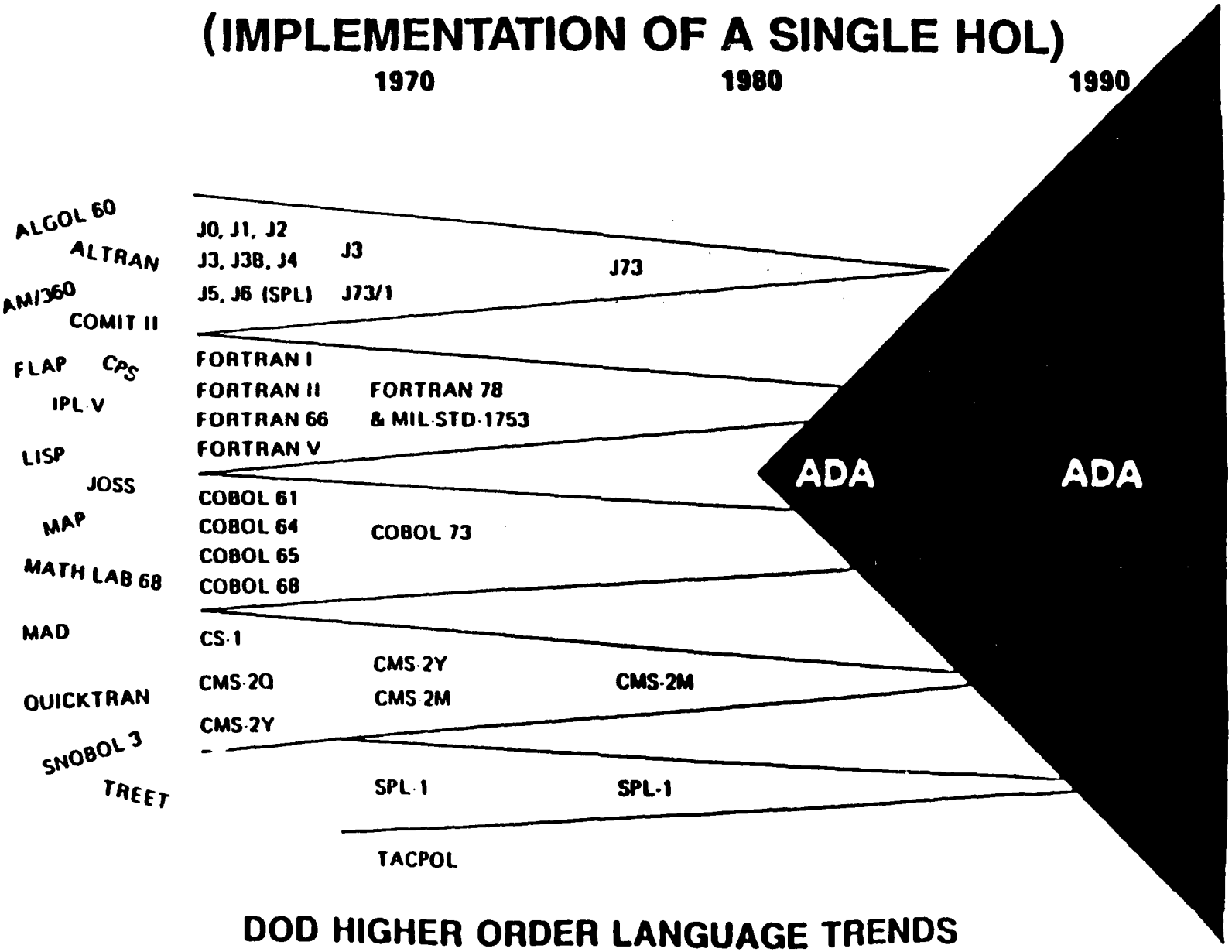
The SEI is charged with getting state-of-the-art software technology -- including but not limited to Ada and STARS results -- into widespread use in developing and suporting DOD software. I chair the Joint Advisory Committee Executive Group for guiding the SEI. I am very impressed with the SEI's assembly of expertise. SEI software experts are working on a variety of initial projects to integrate the latest software development technology and help get it used in PDSS centers and on weapons systems development programs. Their aim is to get the best, new know-how into use soon, so that we can make changes in operational code quickly, without duplication, and without needless rework.

I have a couple of detailed charts on each of these programs. These bring out some technical points of interest for each one. Now let me highlight some of these details. (I'm going to let you read the charts, and as you do, I'll cue your attention to several points that I don't want you to miss.)

DOD SOFTWARE INITIATIVE

ADA:	DOD STANDARD LANGUAGE (MIL-STD-1815)	COMMON BASE FOR DEVELOPING AND SUPPORTING DOD SOFTWARE
STARS:	SOFTWARE TECHNOLOGY FOR ADAPTABLE, RELIABLE SYSTEMS	USES ADA TO INVENT NEW TECHNIQUES TO DEVELOP & SUPPORT SOFTWARE
SEI:	SOFTWARE ENGINEERING INSTITUTE	ACCELERATES THE USE OF MODERN SOFTWARE ENGINEERING TECHNIQUES

TECHNOLOGY TRANSITION
(IMPLEMENTATION OF A SINGLE HOL)



DOD HIGHER ORDER LANGUAGE TRENDS

DOD COMMON PROGRAMMING LANGUAGE -- ADA (PICTURE)

You can tell how important Ada is to us by the fancy art work on this chart! I don't want you to miss the fine point, however: Ada MAKES PROGRAMS EASIER TO MAINTAIN! My experts tell me that this is because Ada enables a sophisticated, modular parts approach. We know what "swap-out/drop-in parts transparency" means for hardware support. Ada will help us get software to the same point.

DOD COMMON PROGRAMMING LANGUAGE--ADA

- ADA IS THE NEW STANDARD DOD COMPUTER PROGRAMMING LANGUAGE
- IMPROVES SOFTWARE AND ELIMINATES DUPLICATIVE COSTS
- SINGLE LANGUAGE REPLACES NUMEROUS OLDER LANGUAGES IN MISSION CRITICAL COMPUTER SYSTEMS INCREASING EFFICIENCY AND MAINTAINABILITY
- FY87: CONTINUES ADA PROGRAMMING SUPPORT TOOLS, ENHANCED COMPUTER VALIDATION TECHNIQUES, INTERNATIONAL STANDARDIZATION EFFORTS

STARS PROGRAM

As I've said, STARS is run by OSD, and I don't pretend expertise in all of STARS' plans. However, this chart just covers three of the major STARS objectives. These are developing:

(a) Reusable parts that make software upgrades easier, and more reliable;

(b) Automated development environments that reflect INNOVATIVE, COOPERATIVE, INDUSTRY LEADERSHIP through identical "*competing primes*" contracts;

(c) Demonstrations of the productivity increases you can get by using Ada.

STARS is just now allocating funds for the latter objective. You may see our SPOs requesting offerings for demonstration of increased productivity with Ada late this year.

STARS PROGRAM

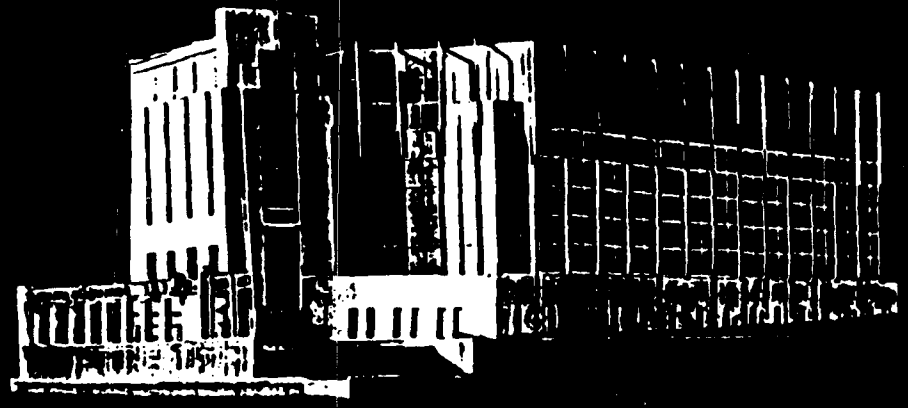
- SOFTWARE UPGRADES
  - PROTOTYPES OF REUSABLE ADA PARTS
- SOFTWARE SUPPORT ENVIRONMENTS
  - INTEGRATED ADA TOOLSETS, TAILORABLE FOR SPECIFIC APPLICATIONS
  - INNOVATIVE "COMPETING PRIMES" CONTRACTS
- PRODUCTIVITY DEMONSTRATIONS
  - "SHADOW PROJECTS" PRODUCE USABLE ADA APPLICATIONS
  - AS PRODUCTIVITY BENCHMARKS

DOD SOFTWARE ENGINEERING INSTITUTE (SEI) (PICTURE)

Just two years ago, the SEI was still a dream. Its the first Federally Funded Research and Development Center (FFRDC) that DOD has established in twenty years. Now it is a functional reality, and by late spring members of the SEI technical staff will be moving into the permanent facility whose artist's conception is shown on this chart. Their efforts will evaluate software engineering technology, and integrate know-how to accelerate the use of advanced techniques on DOD programs.

DOD SOFTWARE ENGINEERING INSTITUTE (SEI)

SOFTWARE TECHNOLOGY TRANSITION



- FEDERALLY FUNDED RESEARCH AND DEVELOPMENT CENTER OPERATED BY CARNEGIE MELLON UNIVERSITY, PITTSBURG, PA
- UPGRADES PROTOTYPE SOFTWARE PRODUCTS TO PRODUCTION QUALITY FOR COMMON SERVICE AND DEFENSE AGENCY USE THROUGHOUT ENTIRE DEVELOPMENT CYCLE
- ACCELERATES TRANSITION OF MODERN SOFTWARE TECHNOLOGY INTO DOD WEAPON SYSTEMS
- FY87: CONTINUES INSERTING SOFTWARE TOOLS INTO THE DEVELOPMENT LIFE CYCLE

SEI PROGRAM

Three aspects of the SEI program are especially relevant to Post Deployment Software Support.

The first result of the SEI PDSS project will be a demonstration of a prototype data base application for keeping software maintenance Tech Orders up to date with the actual changes in F-16 flight code. This work is being done in concert with the folks at Ogden's ALC, including Rick Holsman, who is here today.

The problem of data rights which, when not properly acquired, constrain responsive, and competitive software modifications is being addressed by the SEI's technical and legal group. This group is also advising the Defense Acquisition Review (DAR) Council with its efforts to update the software aspects of the Federal Acquisition Regulations (FARs). So far they've published some sound reports, and helped to crystalize and focus technical and legal experts on this important problem.

You should read this last bullet as "implementing modern software engineering techniques". The SEI will soon release a handbook to help program managers decide when and how best to introduce Ada to their problems. This work, and earlier and continuing efforts to define evaluation criteria for software tools, are valuable in the PDSS world as well as in the world of SPOs and IOCs.

DOD SOFTWARE INITIATIVE

ADA – "DOD WILL SOON ISSUE A DIRECTIVE REQUIRING ITS USE THROUGHOUT THE DOD" SECDEF, 19 NOV 86

STARS – "COMPETING PRIMES" CONTRACTS AWARDS PLANNED FOR '87

SEI – OVER 15 PROJECTS UNDER WAY TO TRANSITION TECHNOLOGY, SUPPORT PROGRAMS, PROVIDE SOFTWARE ENGINEERING EDUCATION AND RESEARCH

A PDSS CHALLENGE
(VHSIC STANDARDIZATION CHALLENGE)

If the Department of Defense and industry are serious about reducing the costs of Post Deployment Software Support, we must not make the same mistakes twice. We know that PDSS costs exceed development costs by a factor of five to one. We also know that the resusability and transportability of mature software would significantly reduce PDSS costs. More importantly, we know that software quality improves with time. We have examined software initiatives to reduce software support costs and increase software quality. However, we would be remiss if we do not also stimulate hardware initiatives that can make an enormous contribution to ensuring software quality and reducing software support costs.

The protocols and instruction set architectures for future VHSIC processors are now being developed for computer technologies which will dominate the embedded computer processor industry for a generation. These protocols and standards enhance or inhibit software transportability, as well as interprocessor communication and interoperability. I specifically challenge the semiconductor industry to cooperate and develop standard DOD VHSIC processors with transparent upgrade capability, and true interoperability that allow the transportation and reuse of our mature applications software. Such an initiative, allowing the transport and reuse of high quality software across DOD embedded processors, could significantly drive down software suport costs for the DOD and reduce IR&D costs for the semiconductor industry.

A PDSS CHALLENGE

- DOD SUPPORTS A COMPETITIVE U.S. SEMI-CONDUCTOR INDUSTRY
- DOD REQUIREMENT - INTEROPERABLE INTEGRATED CIRCUITS
- CHALLENGE - DEVELOPMENT OF STANDARDIZED VHSIC PROCESSORS FOR INTEROPERABLE DOD USE

DOD SOFTWARE INITIATIVE (SUMMARY OF)

This last chart on the DOD Software Initiative calls your attention to the "headlines" on each of these programs:

I participated in the Ada Expo in West Virginia this past November. Mr. Weinberger's endorsement of Ada was unmistakable. I believe very strongly that smart standardization has a big payoff in the post deployment area. This is where Ada will give us the greatest returns, ... on each new program in which we put Ada to use. There's a very strong commitment from the top to make this happen soon and across the board.

STARS aims to fund industry to prototype its best ideas for automating the Ada development process, and for building Ada software parts which speed and simplify the software process. The headline on STARS is the OSD schedule for releasing the single major STARS RFP this year. (I am not up to speed on all of its technical details, but I am assured that the STARS Director is -- he can be reached at the STARS JPO (202-694-0210). Let me just say that his effort promises to fund the innovative integration of a wealth of new ideas for software development, and to involve (through multiple awards and a high degree of subcontracting) a large number of the best systems and the best technology companies who provide software and mission systems for the DOD.

The SEI headlines are really two:

(1) The permanent facility will be fully operational late this year but their efforts for software engineering technology transition, program support, research and education are being worked now by over 100 technical staff members.

(2) Their efforts to formalize a graduate curriculum for software engineering has already crystalized cooperative academic definition of a Masters degree program. Its modules are now being taught at several institutions on a trial basis.

WRAP:

Perhaps this last is the most practical benefit of the DOD Software Initiative to the PDSS world in the very near term: Shortly a new crop of engineers should be graduating and looking towards solving lifecycle software problems on your programs and in your support centers.

SUMMARY

- SOFTWARE ISSUES ARE BEING ADDRESSED
- PERSONNEL ISSUES WILL CONTINUE TO PLAGUE PDSS
- PDSS WORKSHOP INITIATIVES – IMPROVE POSTURE

SECTION 3

PANEL SUMMARIES

(Intentionally Blank)

PDSS PLANNING DURING DEVELOPMENT
PANEL I
PANEL SUMMARY

CO-CHAIRS: Bill Egan, Advanced Technology, Inc.
John Holcomb, AFLC

Without adequate planning, supported by both policy and budget provisions, effective and timely PDSS of MCCR cannot be achieved.

ISSUES:

- Costs and level of resources needed to support the system throughout its life cycle can only be estimated during concept exploration and are constantly revised as system development progresses.

- Policies are not executed correctly because of the lack of education of the implementors.

- Current DOD and Service policies do not adequately address the importance and cost impacts of software on the total system.

- Software rights in data is not adequately addressed in the Defense Federal Acquisition Regulation Supplement (DFARS).

- Documentation requirements for software development and support issues in the DFARS is cumbersome and confusing.

- Services are unsure which directive to follow in acquiring the support software for computer resources in order to perform PDSS (i.e., Information Systems Directive [data processing] or Defense System Directive [tactical]).

- Joint programs do not require joint Service participation in planning PDSS.

- Policy and guidance on the use of DOD-STD-2167 does not emphasize that it should be tailored.

- Program Managers (PMs) do not develop cost effective PDSS plans.

RECOMMENDATIONS:

- Include specific MCCR questions in the Defense Acquisition Board (DAB) major milestone review process.

- Include PDSS requirements in the tactical program development Request for Proposal (RFP).

- Assign software engineering consultants to the Computer Resources Working Group (CRWG).

- Identify the PM as the individual responsible for total life cycle computer resources cost assessment and control.

- Provide a point of contact for users' questions on directives and instructions.

- Disseminate planning information through teleconferences, videotapes, and newsletters.

- Establish MCCR policy for each Service, similar to that being implemented by AFR 800-14 and OPNAVINST 5200.28 with the following modifications:

  -- Identify the software support concept by Milestone II or before preparing the RFP for the development contract.

  -- Select the support concept based on total life cycle costs (in joint programs, the lead Service must consider the optimum balanced approach).

  -- Reflect support requirements (modifiability, licensing provisions, support software) in the development contract.

- Review Naval Air Systems Command (NAVAIRSYSCOM) INST 5230.9 for Service-wide applicability concerning:

  -- Early establishment of software support facilities.

  -- Management of support laboratory assets.

  -- Assignment of system software support activity.

- Include a rights in software data clause in the current data rights policies of the DFARS to obtain unlimited rights to software.

- Require separate formal software acquisition documents.

- Include all software required to support Mission Critical Computer Systems (MCCS) within the MCCS acquisition policies.

- Update MIL-STD-881 to require that software and associated activities/products be identified to provide visibility, cost and schedule, status accounting, and monitoring.

- Emphasize the need for tailoring DOD-STD-2167.

- Develop guidelines for the PM regarding life cycle support implications of nondevelopment items or commercial off-the-shelf resources.

- Encourage PDSS cost collection for both hardware and software.

- Establish a "BOLD STROKE"-like program in all Services to educate commanders in MCCR issues.

(Intentionally Blank)

FORECASTING PDSS RESOURCE REQUIREMENTS
PANEL II
PANEL SUMMARY

CO-CHAIRS: Bernie Price, USA CECOM
Jerry Raveling, Unisys

Successful planning for software resources in support of MCCR requires proper tools to help make decisions. Techniques, with high levels of management confidence and support, must be developed to permit accurate resource forecasting and budgeting for software support activities.

ISSUES:

- PDSS forecasting methods range from "best guesses" to highly complex, automated, computational techniques.

- Standard forecasting models do not exist across the Services.

- Required model characteristics are not clearly defined.

- Model criteria and data definitions are not consistent.

- Requirements for further investigation and research are needed to evolve technology in software cost estimating.

RECOMMENDATIONS:

- For each Service, establish a policy and implementing mechanism directing a Constructive Costs Model (COCOMO)-like forecasting method.

- Establish a standard software data collection initiative with standard data definitions.

- Implement a management and technically based Software Cost Estimating methodology training program.

- Establish a Service oriented research program to insert new and evolving technology in Software Cost Estimating (SCE).

- Support the adoption of a standard DOD SCE model as a long term goal.

(Intentionally Blank)

SOFTWARE CHANGE PROCESS
PANEL III
PANEL SUMMARIES

PANEL IIIA - PDSS MODELING/SUPPORT STRATEGIES
PANEL IIIB - CONFIGURATION MANAGEMENT

PANEL IIIA - PDSS MODELING/SUPPORT STRATEGIES.

CO-CHAIRS: Ron Pruiett, LtCol, USMC, MCTSSA
Owen McOmber, Comptek Research

A review of the Orlando I PDSS model concluded that it:

(1) Was too complex to be adopted as a general process model at the DOD level.

(2) Failed to address the relationship between initial software development and PDSS.

(3) Failed to emphasize the unique set of activities that distinguish PDSS from initial software development.

ISSUES:

- PDSS is not formally defined.

- A simplified joint Service PDSS model is needed.

- Lack of a joint Service PDSS model to support immediate changes.

- Support software strategies are not considered during the Computer Resources Life Cycle Management Plan (CRLCMP) process.

RECOMMENDATIONS:

- Adopt the Orlando I definition of PDSS.

- Identify PDSS activities as three major functions: management, technical, and support.

- Refine and adopt the proposed standard software support process model.

- Incorporate the mandate for management control of the PDSS process in all planning documents.

- Reflect the PDSS strategy decision in the CRLCMP.

(Intentionally Blank)

PANEL IIIB - CONFIGURATION MANAGEMENT.

CO-CHAIRS: Ron Pruiett, LtCol, MCTSSA
Owen McOmber, Comptek Research

Software configuration management (CM) is a critical support function that has the potential for significant cost avoidance if effective and consistent policy directives, implementing standards and common automated software tools are utilized by the Services and industry.

ISSUES:

- Inconsistencies and deficiencies exist in the DOD CM directives and standards as they relate to PDSS activities.

- Existing DOD CM directives and standards are not current (issued in the early 1970's) and are inconsistent with DOD-STD-2167, the approved DOD standard for the development of weapons systems software.

- Service implementations of DOD software CM guidelines, procedures, and practices are inconsistent and incompatible.

- Large numbers of independently developed and maintained, (but functionally equivalent), automated software Configuration Status Accounting (CSA) systems used by the Services, greatly increase overall DOD software life cycle maintenance and training costs.

- Incompatible software CSA tools inhibit the exchange of CSA data among user sites and prevent cost effective transfers of critical CSA data between DOD development and PDSS activities.

RECOMMENDATIONS:

- Initiate a major update of the DOD Configuration Management Plan (DCMP), including the update, rewriting, or replacement of existing related directives, standards, and procedures.

- Develop a common automated software CSA system using the guidelines for writing a SOW and specifications for essential common CSA data elements developed in Orlando II.

- Develop a formal handbook to assist DOD activities involved in the development of automated software CSA systems.

(Intentionally Blank)

PDSS STANDARDS
PANEL IV
PANEL SUMMARY

CO-CHAIRS: Stan Packer, AFLC/OO-ALC
Vern Parsley, CSC

DOD-STD-2167 and DOD-STD-2168 (draft) were developed to be used in an MCCR acquisition and development environment. These standards need to be reviewed from a PDSS perspective.

ISSUES:

- Identify changes to DOD-STD-2167 and DOD-STD-2168 (draft) to incorporate PDSS consideration.

- Identify which requirements of DOD-STD-1467(AR) to incorporate in DOD-STD-2167.

RECOMMENDATIONS:

- Describe the post deployment phase in DOD-STD-2167.

- Define the preliminary software development activities in DOD-STD-2167.

- Address modification to other than DOD-STD-2167 developed items within a DOD-STD-2167 environment.

- Change DOD-STD-2167 title to: "Defense System Software Development and Support".

- Incorporate identified items from DOD-STD-1467 into DOD-STD-2167.

- Incorporate identified items from DOD-STD-1467 Data Item Descriptions (DIDs) into DOD-STD-2167 DIDs.

- Incorporate changes identified by subpanel reviews into DOD-STD-2167.

- Incorporate specified changes to emphasize the software build process.

- Add transition information to the Computer Resources Integrated Support Document (CRISD) DID.

- Provide a means for delivery of documentation for commercially available software in DOD-STD-2167.

- Apply DOD-STD-2167 PDSS changes recommended by panel.

(Intentionally Blank)

PDSS MANAGEMENT INDICATORS AND QUALITY METRICS
PANEL V
PANEL SUMMARY

CO-CHAIRS: Gene Long, AFLC
Jim Miller, SAIC

Management indicators and quality metrics are essential if the DOD and its industry partners are to turn the current DOD-perceived state of software "witchcraft" into a science and to assure continuation of a quality product during PDSS.

ISSUES:

- Lack of quantitative metrics and indicators prevents definition of software quality attributes such as mission effectiveness, availability, and maintainability.

- Lack of indicators and metrics precludes weapon system warranties and effective software risk management techniques.

- Lack of indicators and metrics affects development and support of quality products within performance, cost, and schedule constraints.

- Lack of communication and coordination across the DOD and industry significantly retards the sharing and use of valuable engineering metric disciplines.

RECOMMENDATIONS:

- Establish a full time, Joint Service subgroup of the JLC JPCG-CRM to develop and oversee a management indicators and quality metrics program to:

  -- Build upon the current Air Force (800 series) initiatives.

  -- Incorporate other Service efforts.

- Establish a CRM Subgroup on metrics.

- Automate the metric gathering process to provide consistency, accuracy, completeness, and cost effectiveness.

- Incorporate metrics and indicators into current DOD policy directives and standards.

- Share common indicators, metric tool sets, and data banks across DOD agencies for cost effectiveness.

- Develop a metrics information repository and distribution center/clearing house to promote industry and DOD cooperation.

- Promote research to assure that metrics are kept current with ever changing computer and software technologies.

- Transition metrics and tools from development agencies to avoid redundancy and excessive maintenance costs.

HUMAN RESOURCES IN PDSS
PANEL VI
PANEL SUMMARY

CO-CHAIRS: Linda Doldt, AFLC
Terry Brim, TRW

Actions must be defined to ensure the recruitment, retention, and training of knowledgeable software personnel to support PDSS.

ISSUES:

- New career management procedures are required.

- Educational and training initiatives are necessary.

RECOMMENDATIONS:

- Establish a new software engineering job series (GS-8XX) for the civilian work force.

- Adopt alternative position classification and pay systems for critical PDSS skills (i.e. "pay banding").

- Refine and market a model for computer engineering/ software engineering curriculum.

- Task an ad hoc group to:

  -- Define a consolidated approach to software engineering training.

  -- Create awareness in DOD management of software training and funding requirements.

  -- Assess available training and Service needs.

  -- Develop an automated data base.

- Protect existing manning levels by "fencing off" critical PDSS spaces.

(Intentionally Blank)

SOFTWARE TECHNOLOGY TRANSITION
PANEL VII
PANEL SUMMARY

CO-CHAIRS: Myron Holinko, USA CECOM
John Marciniak, Marciniak and Associates

The panel's objective was to identify policies and methods for transitioning necessary software tools while controlling their proliferation so that PDSS needs are met in a cost effective manner.

ISSUES:

- No DOD level policy exists to explicitly address PDSS within the system development life cycle.

- No uniform DOD policy is currently used in contracting for support software.

- DOD and contract program managers who are developing systems with PDSS requirements do not thoroughly understand the software development process, life cycle, and impact of supportability issues on the final products.

- Existing proliferation of software support environments and similar tools throughout DOD.

- Due to problematic DOD data rights policies, which convert highly valued technology to public domain, contractors are unwilling to use their state-of-the-art tools and capabilities in development of DOD systems.

- Identification, procurement, and widespread distribution of common PDSS tools, methods, and processes is inhibited by:

  -- Separation of the Services.

  -- Organizational and command separation within each Service.

  -- Alignment of PDSS organizations along acquisition program lines.

  -- Concentration on immediate operational problems.

- Ada productivity improvements will not be realized by PDSS activities in the near future.

RECOMMENDATIONS:

- Update DODD 5000.29 and its implementing instructions to strengthen the OSD oversight for software development, support decisions, and PDSS consideration during the acquisition process.

- Review and modify DOD-STD-1467 to include PDSS technology transition requirements.

- Develop a PDSS training program for PMs.

- Use DOD-STD-1838 (draft) Common Ada Programming Support Environment (APSE) Interface Set (CAIS).

- Develop or modify the DOD Acquisition Regulations (AR) so that state-of-the-art tools are available for PDSS.

- Establish a joint Service PDSS software commonality office at the command level to assess and distribute software tools, support users, and provide intraservice coordination.

- Pursue an other-than-Ada PDSS technology improvement program for pre-Ada systems, including increased tasking to the Software Technology for Adaptable and Reliable Systems (STARS) program and the Software Engineering Institute (SEI).

- Establish cost effectiveness criteria for Ada conversion.

MCCR SECURITY
PANEL VIII
PANEL SUMMARY

CO-CHAIRS: Sharon Muzik, NESEA
Robert Converse, CSC

The PDSS crisis is exacerbated by the lack of computer security in delivered systems. Retrofitting security into existing systems is costly and marginally effective.

ISSUES:

- Insufficient guidance for specifying and assessing MCCR security requirements.

- Lack of clear guidance for implementing and identifying MCCR security requirements.

- Inadequate capabilities for evaluating and certifying MCCR systems.

- Existing computer security R&D program does not adequately address MCCR requirements.

RECOMMENDATIONS:

- Embed computer security requirements in DOD-STD-2167.

- Develop a computer security implementation guidebook.

- Establish organic Service certification and evaluation capability.

- Develop better guidance on identifying security requirements.

- Support an R&D program to:

  -- Adapt existing software engineering tools to enhance capabilities of computer security requirements in new systems and identify computer security weaknesses in existing systems.

  -- Develop automated tools and techniques to support trusted systems in the future.

  -- Develop efficient and effective MCCR security architecture.

(Intentionally Blank)

SECTION 4

PANEL PROCEEDINGS

(Intentionally Blank)

PDSS PLANNING DURING DEVELOPMENT
PANEL I
PROCEEDINGS

OBJECTIVE.

Panel objective was to identify activities of MCCR software support activities that must be planned for during system development.

BACKGROUND.

Proper planning is necessary to enable efficient, effective software support after the developed software is deployed to the user. Software designs must consider the chosen support concept to facilitate the separation of software support responsibilities (e.g., Government, contractor, user). Software support tools, associated equipment, and facilities must be acquired in a timely fashion to permit the acceptance of support responsibilities by designated organizations. Appropriate and timely budgets must be established in order that effective PDSS takes place. This has frequently not been done in the acquisition of MCCR software. Therefore, it is imperative that planning for support be performed during the development phase of MCCR software.

SCOPE.

The central theme of Orlando II was "Solving the PDSS Challenge." The workshop addressed various aspects of PDSS to identify areas that offer significant payoffs in terms of cost reduction, improved system reliability, streamlining of the PDSS budgeting process, and, most importantly to Panel I, effective planning and management. The challenge for Panel I, as discussed in the Orlando II Master Plan, was described as:

a. Identify, define, and prioritize PDSS activities that must be planned for during the software development phase.

b. Identify changes to current DOD regulations, standards, and directives to implement each aspect of planning identified above.

c. Identify methods of streamlining the budgeting process so necessary software support resources are provided at the time of system deployment.

ASSUMPTIONS.

In planning the approach of Panel I, the following basic assumptions were made:

a. Planning, policy, and budgeting recommendations will be limited to PDSS issues only.

b. Planning, policy, and budgeting recommendations will be addressed, wherever possible, at the Service level.

c. Recommendations must be implementable and compatible with the charter of the of JLC JPCG-CRM Joint Policy Coordinating Group.

d. The JLC JPCG-CRM does not make joint Service policy.

e. Policy recommendations will not include changes to DOD-STD-2167 and DOD-STD-2168.

APPROACH.

Subsequent to the opening general session that took place on Monday morning, Panel I met as a group in their assigned room for the first time.

The purpose of this initial group session was to:

a. Review the Panel's objectives and intended products.

b. Identify and review the general approach to the Panel's operation, schedule, administration, and relation to overall workshop goals and objectives.

c. Identify and review planned approach to subpanel, group and joint sessions.

d. Permit the co-chairs and each panel member to introduce themselves to the Panel members.

PANEL SESSIONS.

Panel I began its deliberations by receiving several briefings structured to provide a framework for its recommendations. These briefings were provided by both Panel members and invited guest speakers. Individuals were contacted prior to the workshop and requested to present briefings on selected issues and activities related to planning, policy, and budget concerns that must be considered during the development phase of the system life cycle. Briefings were from 30-45 minutes in length or longer. A list of briefings is included.

a. Ms. Paula Davis of the Information Systems Division, Office of the Chief of Naval Operations, outlined OPNAVINST 5200.28, the Navy's new policy on life cycle management of mission critical systems. She identified the scope of the policy by specifying the types of resources that fall within each of the

research, development, and acquisition processes. She presented the rationale for developing a new Navy policy on software, and discussed several factors that need to be considered when planning system requirements. Additionally, she addressed the Navy policy on standardization. Among the facets of this policy that she covered were: the role of the CRLCMP, joint systems, interfaces, rights to computer resources, and computer resource management.

b. Mr. A.T. (Tom) Smith of the Naval Air Systems Command (NAVAIRSYSCOM) presented a briefing on NAVAIR Instruction (NAVAIRINST) 5230.9, "Tactical Embedded Computer Resources Policy in the Naval Air System Command". He began by discussing the NAVAIR experience in developing the policy. The formulation of the policy was evolutionary and predicated upon considerable input from various field activities. A key point in implementing the policy was educating management on why the proposed policy is important. The importance of its organizational structure and accomplishment of its precepts will be essentially meaningless unless there is a well organized, adequately staffed group to ensure its implementation. Mr. Smith then outlined the precepts of NAVAIRINST 5230.9. The policy covers both airborne computer/ software and related hardware. PDSS planning early in the program was emphasized and the System Software Support Activity (SSSA) structure utilized to perform support was discussed.

c. Mr. Bruce Baxter, head of the Computer Resources Division at the Pacific Missile Test Center (PMTC), presented his perspective on the views and concerns of a MCCR software life cycle support activity. Early PDSS planning was emphasized and the SSSA support structure discussed. He pointed out that while significant progress has been made relative to PDSS planning and budgeting during development, his activity viewed with concern that not enough was being done to accommodate the embedded software now in, or soon entering, initial service life.

d. Colonel Jerry Stewart, Commanding Officer, Marine Corps Tactical Software Support Activity (MCTSSA), Camp Pendleton, California, gave an overview of the PDSS activities within his command. He indicated that MCTSSA is currently arranging to be the post deployment support activity for some joint Service programs.

e. Captain Rich Armour, HQ, USAF, briefed the panel on the newly revised version of Air Force Regulation (AFR) 800-14. After giving the panel a background of the policy formulation process, he outlined the major policy issues. These included earlier support planning, increased user/supporter involvement in acquisition and support planning, sanctioned user support of mission software, and phased development of the CRLCMP. The regulation recognizes that no one single support concept is universally optimum for all systems, and it describes several

support concepts. AFR 800-14 also sets forth improved standards and commercial-off-the-shelf (COTS) policies. Finally, the regulation decreases the importance of the transition phase between introduction of a system into the operating environment and program management responsibility turnover.

f. The Army perspective on PDSS was offered by Frank Sisti of VITRO Corp.; Colonel Reed, Communications and Electronics Command (CECOM), Ft. Monmouth; and Jack Byers, Army Materiel Command (AMC). Mr. Sisti began by providing an overview of the responsibilities of AMC, the Training and Doctrine Command (TRADOC), and the Information Systems Command (ISC). Colonel Reed then discussed tactical, sustaining, and strategic spheres of the Information Management Area. He also discussed the role played by CECOM. Jack Byers rounded out the briefing by describing the role that the AMC plays in software life cycle support. It was noted that the Army currently does not have an Army-wide regulation covering life cycle software support. Rather, subordinate commands have developed their own regulations and manuals.

g. Mr. Chuck Gordon of CACI then presented an overview of DOD-STD-1467, "Military Standard Software Support Environment," which is applicable to the Army. The objective of this standard is not to specify a standard support environment but rather to ensure complete life cycle support capability. The Army's approach is to establish an organic support capability where possible. Accordingly, this standard attempts to ensure compatibility between the development and support environment. It emphasizes the importance of early PDSS decision making. For example, in the context of data rights in support technology, the standard indicates that RFPs should require contractors to identify in their proposals any intended use of proprietary products during software development.

h. Ms. Anne Martin of the Software Licensing Project of the SEI gave a presentation on the project's recent study of data rights issues arising in software life cycle support. The first phase of the study entailed an examination of the DOD environment in which software support planning and performance is performed. This involved ascertaining the technological needs required to perform PDSS and the variables impacting those needs. The project concluded that the sophisticated nature of PDSS requires a transition of system expertise from the developer to support personnel. Because this expertise is embodied in technology that often incorporates valuable proprietary information, it is in the PDSS context that the clash between the needs of DOD and the proprietary interests of industry are magnified. A resolution to this clash, requires early PDSS planning to accurately identify the technology required for PDSS as well as the degree of access needed in that technology. Because the need to transition technology may vary based on several managerial and technical

factors, good PDSS planning requires consultation with user and support personnel. In order for PMs to be able to structure an acquisition to acquire needed support technology, they need to be able to employ a variety of legal methodologies to access that technology. This calls for a flexible software acquisition policy that balances the needs of DOD with the proprietary interests of industry. The project hopes to be involved in the development of such a policy.

SUBPANEL SESSIONS.

On Wednesday the panel broke into subpanels IA, IB, and IC. Tasks, goals, and members of each subpanels were:

SUBPANEL IA

Task: Identify, define and prioritize PDSS activities that must be planned for during the software development phase.

Goal: List of essential and recommended PDSS planning activities.

Members:
John Holcomb < Group Leader >
Russ Edgerton
Don Frogner
Richard Healy
Ken Lee
Paul Mauro
Jerry Stewart
Bud Wasgatt

SUBPANEL IB

Task: Identify changes to current DOD regulations, standards, and directives to implement each aspect of planning necessary.

Goal: Recommend specific modifications to DOD standards, directives, and regulations to implement each planning activity.

Members:
Frank Sisti < Group Leader >
Donna Cover
Paula Davis
Anne Martin
Linda Sanders
Frances Soskins
Marilyn Stewart
Bill Spaulding
Don Zana

SUBPANEL IC

Task: Identify methods of streamlining the budgeting process so that necessary software support resources are provided at the time of systems development.

Goal: Develop recommendations to improve the budgeting process.

Members:		
	Bill Egan	< Group Leader >
	Tom Smith	< Guest Speaker >
	Rich Armour	
	Bruce Baxter	
	Jack Byers	
	Cenap Dada	
	Kevin Porter	
	Paul Sonnenblick	

Wherever possible, members were allowed to select the panel and topic area of their choice. Subpanel leaders and recorders were selected by the subpanel members in cooperation with the panel co-chairs.

Subpanels discussed their assigned topic areas, identified planned recommendations in accordance with the panel outline, and prepared written notes on major items of discussion and decisions.

Issues of a general nature, or those that may have the potential for impacting another subpanel or panel, were identified by the subpanel leader and reported to one or both panel co-chairs. The co-chairs facilitated and coordinated required panel or subpanel interaction.

FINAL GROUP SESSION.

The subpanels reformed for a Group Session in the afternoon on Thursday. Each subpanel leader or recorder provided a review of the subpanel's deliberations, and reported on the recommendations developed by the subpanel.

Based on these subpanel reports, the co-chairs prepared a panel summary for presentation at the following Joint Session.

FINAL WORKSHOP JOINT SESSION - FRIDAY.

For the final workshop Joint Session, each of the panels prepared and presented a 20-30 minute briefing. The briefing summarized the panel's work during the week, provided a review of the panel's recommendations to the JLC JPCG-CRM and provided other germane and salient information.

PRELIMINARY WRITTEN REPORT.

Panel co-chairs and panel participants prepared and submitted to the JLC Workshop Committee a preliminary written report, based upon draft written material prepared at the workshop and on the final Joint Session briefing.

FINAL WRITTEN REPORT.

Panel co-chairs, subpanel leaders and recorders developed, prepared, reviewed, coordinated, and issued a final panel report to the JLC Workshop Committee in accordance with the JLC's and the panel's schedule. These reports present details on the panel's charter, deliberations, and recommendations. Products were structured in accordance with the following.

a. Product Number. Each specific panel product should be listed separately with descriptive title.

b. Priority. List products in order of their importance as needed to support the operational requirements of the MCCR.

c. Required Actions. Follow the title with an elaborating paragraph that states definitively the action required. List the cogent factors where applicable such as:

1. Near Term (0-1 year), Mid Term (2-4 years), and Long Term (5-10 years) solutions.

2. Return on investment (short, medium, long term). Include an estimated cost and time to implement.

3. Dependencies upon other actions or recommendations.

4. List alternatives. State the number of panel members concurring with minority opinions, if relevant.

5. Method of implementation.

6. Justification for your prioritization of this product.

7. Detailed product.

PRODUCTS

PRIORITIZED (ESSENTIAL) PLANNING ACTIVITIES.

Increase MCCR Visibility During Reviews - System acquisition processes do not adequately ensure proper life cycle computer resources supportability. The PM's mission and charter are

limited to development responsibilities only and must be expanded to include a total system life cycle perspective. Deficiencies in MCCR acquisition occur as a consequence of insufficient MCCR expertise available to the PM from inception of the system (e.g., poor RFP preparation, no visibility for MCCR in milestone reviews). The Services must increase visibility and accountability for MCCR issues by enhancing the major Milestone review processes by including specific MCCR PDSS related questions. DAB members qualified to assess responses should be present. The request for proposal preparation process must be improved to preclude deficiencies in MCCR acquisition and long-term supportability.

Recommendation 4-1-01 (Mid Term, 2-4 years): Include specific MCCR questions in the DAB major Milestone review process. The April 1981 "Embedded Computer Resources and the DSARC Process" guidebook published 30 April 1981 by the Office of the Under Secretary of Defense for Material Acquisition should be reviewed, updated, and reissued with strong JLC endorsement.

Recommendation 4-1-02 (Mid Term, 2-4 years): Include PDSS requirements in the tactical program development RFP. Guidance needs to be developed to help PMs by providing general and specific PDSS recommendations for inclusion in the RFP for mission critical systems that will include computer resources. The designated life cycle support activity should be an active participant in the RFP preparation process. This should be a requirement across all of the Services. Current Air Force and Navy life cycle management policies now include these provisions.

Recommendation 4-1-03 (Mid Term, 2-4 years): Assign software engineering consultants to the CRWG. Expand the role and responsibilities of PM's CRWG by including trained personnel to provide comprehensive software engineering consultation in the following representative areas:

a. Use and extent of standards, documents, and DIDs commensurate with complexity of system.

b. Feasibility of partitioning system functional requirements between hardware and software.

c. Long Term MCCR supportability requirements (facilities, personnel specialties, support environment requirements).

d. MCCR cost estimates, including cost of any licensing or data rights considerations for Nondevelopment Item/Commercial Off-the-Shelf (NDI/COTS) resources and tools.

e. Capabilities of existing hardware and software suitability for meeting system performance requirements, in order to curtail proliferation of types of MCCR to be supported.

Improve MCCR Cost Estimates - For a successful system, not only development cost, but cost and level of resources needed to support the system throughout its life cycle, must be estimated during concept exploration and must be updated as system development progresses.

Recommendation 4-1-04 (Mid Term, 2-4 years): Identify the PM as the responsible individual for the assessment of total life cycle MCCR costs, and task the PM with the control of MCCR development costs. Current Service life cycle management policies should be revised to include this provision. This would be a low cost but effective action that could be accommodated during normal review cycles of existing policy.

Improve PM Awareness of MCCR Requirements - Many implemented policies are not executed correctly because of the lack of well trained implementors. When clarification is necessary, develop and issue handbooks and implementation guidance in parallel with the policy statement. Whenever possible, augment usual information dissemination techniques through the use of teleconferencing, videotape, and newsletters. Furthermore, provide a point of contact to address users' questions.

Recommendation 4-1-05 (Mid Term, 2-4 years): The JLC JPCG-CRM PDSS Subgroup should initiate a PDSS awareness program to:

a. Provide a point of contact for users' questions on directives and instructions.

b. Disseminate planning information through teleconferences, videotapes, and newsletters.

Improved Planning Policy - Significant improvements have been implemented in both Air Force and Navy MCCR life cycle management policies that strengthen requirements supporting effective planning for PDSS. However, there are areas where improvements to existing Service policy (e.g., AFR 800-14, OPNAVINST 5200.28, etc.) could be implemented.

Recommendation 4-1-06 (Mid Term, 2-4 years): Each Service should include the following PDSS planning policies in their respective life cycle management regulations and instructions. Specific policy should require:

a. Identification of the software support concept by Milestone II or before preparing the RFP for the development contract.

b. Selection of the support concept based on total life cycle costs (in joint programs the lead Service must consider the optimum balanced approach).

c. Reflect support requirements (modifiability, licensing provisions, support software) in the development contract.

Adoption of Existing Working Policy - Over the past decade and even since the advent of Orlando I, there have been extensive and effective PDSS planning activities identified and implemented. In June of 1983 NAVAIRSYSCOM released NAVAIRINST 5230.9 "Policy and Procedures for the Establishment and Operation of Naval Air Systems Command Systems Software Support Activities". This instruction establishes the requirements for NAVAIR life cycle SSSA and gives general policy, procedures, responsibilities, and operating relationships pertaining to their mission, functions, direction, and support. Of particular interest to the Planning PDSS During Development panel were some key provisions relative to early planning activities. It was strongly felt by the panel that all Services could take advantage of years of preparation and coordination by NAVAIR and adopt some of the creative planning initiatives instituted by that command.

Recommendation 4-1-07 (Near Term, 0-1 years): The Services, via the JLC JPCG-CRM PDSS Subgroup, should sponsor a review of NAVAIR policy (NAVAIRINST 5230.9) for applicability concerning:

a. Early establishment of software support facilities.

b. Management of support laboratory assets.

c. Providing to the SSSA a Force Activity Designator (FAD) priority equal to that of the system being supported.

This should prove to be a low-budget activity that could be accomplished easily by the PDSS subgroup. The Navy has already implemented all of these provisions in OPNAVINST 5200.28. All of the provisions were strongly endorsed by Air Force, Army, and Marine Corps Panel I representatives.

PDSS Planning Activities - The Planning PDSS During Development Subpanel determined that there were eight essential planning functions that must be accomplished in synchronization with specific system acquisition milestones. These activities are related to support environment, facilities, personnel, training, plans, procedures, fielded software, and mission equipment, as indicated in Figure 1. The specific operational date for each of these have been determined and are represented by an "O" aligned with the activity's not-to-exceed milestone. These planning activities should be universally adopted by all the Services.

Recommendation 4-1-08 (Near Term, 0-1 years): Develop a PDSS planning guidebook that t es required activities to major development milestones. Figure 1 should be established as the JLC JPCG-CRM PDSS subgroup endorsed basic frame of reference for all PDSS Planning During Development Activities. All Services

should ensure that the not-to-exceed milestone dates identified are reflected in each of their respective life cycle management policies. In concert with this, each Service should also ensure that the following PDSS planning requirements are included in their respective life cycle management policies.

a. Formally designate and task the software activity prior to Milestone I.

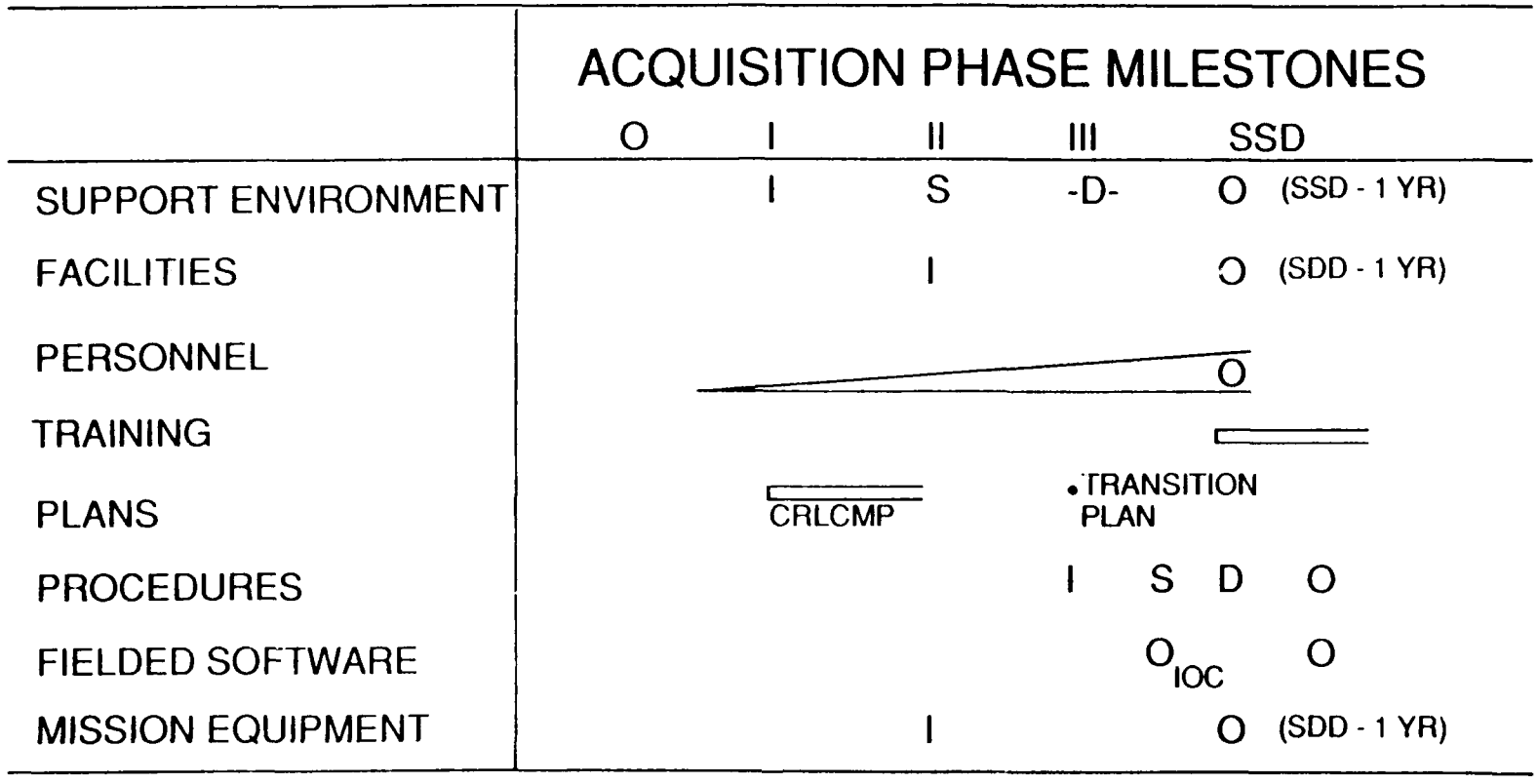
b. The SSSA should be designated as principal in CRLCMP preparation with coordination authorization after Milestone I.

c. The SSSA should be formally tasked to perform or assist in performing IV&V for MCCR software during system acquisition.

Modifications to DOD Standards, Directives, and Regulations Affecting PDSS Planning - DOD and Service level policies must be revised to enhance software visibility in system acquisitions and streamline the acquisition process. Current DOD and Service policies do not adequately address the importance of software in systems and the large impact that software has on systems life cycle costs. Specifically, changes are required as delineated in the following subparagraphs:

Recommendation 4-1-09: The need to perform software support for mission critical defense systems after deployment is not adequately addressed in the current rights in data policies of the Defense Federal Acquisition Regulation Supplemental (DFARS) 52.227-7013. Include a rights in software data clause in the current data rights policies of the DFARS to obtain unlimited rights to software.

Recommendation 4-1-10 (Near Term, 0-1 years): Recommend the DAR Council modify the DFARS to properly reflect the reality of today's software intensive systems by requiring that software development and support issues be separately addressed in formal acquisition documents (e.g., Acquisition Plans and related documents as appropriate.)



LEGEND: I = IDENTIFIED, S = SPECIFIED, D = DEVELOPED, O = OPERATIONAL

FIGURE 1. PDSS Planning Activities

MIL-STD-881A Revision to Address Software - Work break-down structure guidance specified in MIL-STD-881A does not emphasize nor recognize the magnitude of systems software cost. Applying MIL-STD-881A can result in no visibility of software costs, and therefore, the inability of acquirers to track software costs and schedule status (through use of the DOD Standard Cost and Schedule Control System). Changing MIL-STD-881A to address software will result in higher visibility of software in acquisition, and an enhanced ability to manage programs.

Recommendation 4-1-11 (Near Term): Modify MIL-STD-881A to reflect the terminology and methodology of DOD-STD-2167 as well as to require software and associated activities and products to be identified so as to provide visibility and cost and schedule status reporting and monitoring.

Management of Support Computer Resources as an Integral Part of System's Acquisition - Current DOD guidance and regulations are ambiguous with respect to acquisition and management of computer resources for support of mission critical defense systems. Specifically, Services are unclear as to whether to acquire the support computer resources required to perform PDSS (generally commercially available computer resources) under the Information Systems directives (7920 Series) or Defense System directives (5000 Series). The premise and intent of the Warner Amendment with respect to streamlining the acquisition process has not been achieved. The result is that, in some cases, two sets of acquisition policies are followed and two sets of approvals must be obtained. Clean, concise policy, easily understood and readily applied, will result in a drastic reduction of bureaucratic machinations within DOD and the Services.

Recommendation 4-1-12 (Near Term, 0-1 years): - Recommend that the management of all computer resources be included as an integral part of system's acquisitions. Include all software required in support of MCCR within the MCCR acquisition policies. If it is necessary for DOD to have two sets of requisition policies, one for defense system (communications, command, control, intelligence weapons, tactical, and strategic) and one for automated information systems (data processing, business, nontactical), then change the computer resources required to perform PDSS as parts of the systems they support for the entire life cycle of the system. Also, review and modify acquisition policies to incorporate the discipline of the development and production process for post Milestone III software activities.

Policy to Require Computer Resource Joint Service Participation on Joint Programs - Regulations on joint programs do not require joint Service participation in planning PDSS nor do they provide guidance on funding and cost sharing for post-deployment support. Early joint planning could reduce software support costs if concepts such as centralized software support have been analyzed.

Recommendation 4-1-13 (Near Term): Require computer resource joint Service participation on joint programs. Services should incorporate a statement similar to the Navy policy in OPNAVINST 5200.28, paragraph 19, which states:

> "Joint Systems. For allied and joint Service systems in which the Navy is the lead Service, an interservice working group will be established. This group will ensure that analysis is performed to determine the optimum support approach for the life cycle; cost implications of major software support options; and the impact on operational needs, system life cycle costs, compatibility, interoperability, configuration management, and system integration. This group will document this analysis and make recommendations to the Developing Agency concerning the support approach."

Tailoring of DOD-STD-2167 - Service policy and guidance on the use of DOD-STD-2167 do not emphasize that this standard should be tailored to meet the specific program characteristics. Guidance is not available to allow acquirers to contractually require the minimum set of documentation required to organically support mission critical defense systems software. When DOD-STD-2167 is required in its totality or is misapplied, the net result is higher life cycle costs.

Recommendation 4-1-14 (Near Term): Require tailoring of DOD-STD-2167. Services should emphasize the need to tailor the requirements of DOD-STD-2167 in order to allow for the cost-effective acquisition of systems while balancing the cost of acquisition with effective software development and support requirements. Also, JLC should sponsor development of an automated tailoring tool to assist development activities in tailoring DOD-STD-2167.

RECOMMENDATIONS TO IMPROVE THE PDSS BUDGETING PROCESS.

PDSS Funding Structure - As noted in the Orlando I report, funding of embedded software acquisition and support across the Services is provided through a variety of methods, using a mix of operations and maintenance, R&D, procurement, and modification appropriations. The Orlando I report advocated streamlining this funding process (see Orlando I recommendation No. 2) and establishing a separate "funding line" for PDSS (see Orlando I recommendation No. 20). The panel found that the DOD Planning, Programming and Budgeting System (PPBS) is largely driven by Congress, Office of Management and Budget (OMB), Office of the Secretary of Defense (OSD), and the individual Service organizational structures. While PPBS streamlining is desperately needed, pursuing it for embedded software alone would be infeasible and would fragment the funding of total systems.

The use of multiple appropriations has caused some problems in the Air Force where congressional staffers have disallowed the use of aircraft modification funds for hardware related software changes. SSSA managers in the other Services considered the use of multiple appropriations as a source of flexibility rather than a constraint.

Recommendation 4-1-15 (Near Term): The panel found that creation of a new appropriation or program element to fund PDSS would aggravate the fragmentation of system level management, which is a prime consideration during both acquisition and support, and would not necessarily enhance the adequacy or stability of PDSS funding. However, the panel does recommend that the JLC emphasize to Congress, through OSD, the need for consolidating the funding of hardware modification and associated software changes for Air Force aircraft modification programs. In the area of PDSS, the subpanel concluded that in a major percentage of cases, costs are sufficiently projected and tracked by the Services. The subpanel therefore concluded that the Services have taken major steps toward accomplishing recommendations numbers 15 and 18 from the Orlando I workshop. The subpanel concurred that the Services should continue to adopt policy and procedures to ensure that PDSS costs are for systems that are properly projected and identified.

PDSS Software Costing - Two recommendations of Orlando I dealt with the identification of software costs. Recommendations numbers 15 and 18 appear to apply to the total system life cycle, including system development, system modification programs and PDSS. PDSS, in its totality, is a critical element in supporting operational forces by making changes that are the direct result of operational forces trouble reports that have impact on operations readiness or safety and that provide improved system performance. In dealing with software costing, the subpanel divided the issue into two separate categories:

a. System development and modification including both hardware and software.

b. PDSS required to perform changes to tactical applications software programs that are not the result of companion hardware changes.

In the area of system development and modification, the subpanel found that an overly simplistic view seems to pervade. This view holds that simply collecting software costs together with hardware costs would provide sufficient visibility into the development process. For this to be meaningful, the subpanel recognized that cost information must be collected for the other systems disciplines (systems engineering, integration, testings, etc.) as well. Further, the panel concluded that while certain benefits can be derived by collecting software cost information,

it is not always practical to attempt to collect cost for all software configuration items in a modern weapon system.

Recommendation 4-1-16 (Near Term): Encourage PDSS cost collection for both hardware and software. The subpanel concluded that cost collection requirements should allow for the differing levels of hardware and software in the system, specific contractual requirements, and the impact on resources, provided the costs breakout is representative of the total system software costs (e.g., those software configuration items within the weapon system that will subsequently require PDSS). Collection of software cost data will enhance pre and post deployment cost estimating and projections; identification of the reasons for cost growth; identification of future personnel needs; identification of areas to target for productivity improvement; and assessment of the impact of using new tools and standardization techniques.

Software Support R&D (Recommendation No. 27) - Orlando I recommended that the JLC establish the mechanism to establish the need for support tool funding prior to R&D initiation.

Orlando II, Panel IC, after reviewing the titled recommendation, concluded that, since Orlando I, formal mechanisms with the Air Force and Navy have been implemented that establish the support tools required prior to Milestone II of a project. These formal recommendations are put forth in AFR 800-14 and OPNAVINST 5200.28 for the Air Force and Navy respectively.

AFR 800-14, Section 3-1, provides for the acquisition and support planning criteria. More specifically, Chapter 6, Section 9, details that "the dedicated hardware and software necessary to support the system" will be acquired.

OPNAVINST 5200.28, Section 6, Paragraph d, requires that "standard purpose support software and automatic test tools shall be used to the maximum extent." More specifically, Section 7, requires costs associated with post deployment support be identified, budgeted, monitored, and controlled.

Recommendation 4-1-17 (Near Term): Orlando II, Panel IC, recommends that all Services develop and refine policies and instructions as pertains to software support similar to AFR 800-14 and OPNAVINST 5200.28.

Need for PDSS Awareness - During the deliberations of the Planning PDSS During Development panel, a presentation of the USAF Systems Command (AFSC) "BOLD STROKE" awareness briefing was provided by Col. Casper Klucas (HQ AFSC/PLR). Highlights of this briefing were also included in the speech by Major General Monroe T. Smith, AFSC Deputy Chief of Staff for Product Assurance and Acquisition Logistics.

Recommendation 4-1-18 (Near Term): Establish a program similar to "BOLD STROKE" in all Services to educate commanders in MCCR issues. Panel I arrived at a unanimous conclusion that the best way to obtain necessary consideration for PDSS concerns is to make cognizant management aware of the problem. Therefore, Panel I strongly recommends that all the Services develop and implement a program similar to that of the AFSC. "BOLD STROKE" was viewed as a significant and timely activity that just may do more to solve the PDSS challenge than anything else.

(Intentionally Blank)

PANEL I LIST OF BRIEFINGS.

1. "Navy Tactical Software Policy,"
   Paula Davis, OPNAV-945C, (202) 697-7216

2. "Tactical Embedded Computer Resources (TECR) Policy in the Naval Air Systems Command,"
   Tom Smith, NAVAIR, (202) 692-7035

3. "AMC Life cycle Software Engineering,"
   Jack Byers, HQ AMC, (202) 274-9309

4. "Software Licensing Project,"
   Anne Martin, Software Engineering Institute,(412) 268-7622

5. "Budgeting for Software in the Naval Air Systems Command,"
   Tom Smith, NAVAIR, (202) 692-7035

6. "Panel I---PDSS Planning During Development,"
   John Holcomb, OC-ALC/MMECM, (405) 736-5609 and
   Bill Egan, NAVAIR, (202) 746-3775

7. "Electronic Warfare Directorate,"
   Bruce Baxter, Pacific Missile Test Center, (805) 989-9405

8. "Revision of AFR 800-14, Life Cycle Management of Computer Resources in Systems," HQ USAF/SCPX, (202) 695-0756

(Intentionally Blank)

PANEL I BIBLIOGRAPHY.

1. DOD-STD-2167, "Defense System Software Development," 4 June 1986

2. DOD-STD-2168, "Defense System Software Quality Program, (Draft)" 1 August 1986

3. JLC, "Final Report of the JLC Workshop on PDSS for MCCS (Orlando I)," Vol. I - Executive Summary, and Vol. II - Workshop proceedings, June 1986

4. JLC, "Action Plan for Implementing Recommendations made at Orlando I, Workshop," 26 March 1986

5. AR 70-1, "Systems Acquisition Policy and Procedures," 12 November 1986

6. AR 70-XX, "Management of Army Mission Critical Computer Resources," 7 November 1985

7. AFR 800-14, "Management of Computer Resources," 29 September 1986

8. MCO 5200.23, "Management of Embedded Computer Resources in the Marine Corps," August 1982

9. OPNAVINST 5200.28, "Life Cycle Management of Mission Critical Computer Resources (MCCR) for Navy Systems Managed under the Research, Development, and Acquisition (RDA) Process," 25 September 1986

10. NAVAIRINST 5230.9, "Policies and Procedures for the Establishment and Operation of Naval Air Systems Command Systems Software Support Activities," 14 June 1983

11. SEI TECH REPORT CMU/SEI-86-TR-1, "Toward a Reform of the Defense Department Software Acquisition Policy," April 1986

12. DOD-STD-1467, "Software Support Environment," January 1985

13. DOD-STD-881A, "Work Breakdown Structures for Defense Materiel Items." April 1975

(Intentionally Blank)

FORECASTING PDSS RESOURCE REQUIREMENTS
PANEL II
PROCEEDINGS

OBJECTIVE.

The objective of Panel II Forecasting PDSS Resource Requirements was to "identify (if feasible) a standard PDSS forecasting model."

BACKGROUND.

The basis for the establishment of Panel II, Forecasting PDSS Resource Requirements has its roots in the second (Monterey II) and third (Orlando I) JLC software workshops, and in an event which occurred a little over six years ago. In October 1980, the Requirements Committee of the Electronics Industries Association (EIA) presented the results of a year-long study of DOD budgeting for computer hardware and software/services. The study was entitled, "DOD Digital Data Processing Study - A Ten Year Forecast." The hypothesis behind the study was that an ever increasing share of the DOD electronics budget was being earmarked for digital computers and software.

It is a significant understatement to say that the data presented was of extreme interest to the DOD and industry. Highlights of the report were:

- Defense electronics will increase from $20.1B in FY80 to $75.7B in FY90. Defense computers will increase from $6.7B in FY80 to $15.8B in FY90...from 33% of defense electronics in FY80 to 60% in FY90.

- Software and services will increase from $4.6B in FY80 to $37.2B in FY90...from 69% of the total defense computer expenditures in FY80 to 81% by FY90.

While not directly attributable to the EIA 1980 report, the establishment of Panel D - Estimating Software Costs at JLC's Monterey II workshop was in concert with the basic thrusts of the EIA report and a direct precursor to future panel assignments, including ours, for Orlando I & II. Panel D workshop objectives were to (quoting from their panel report): "Evaluate existing software cost estimating models and recommend to the JLC JPCG-CRM a triservice approach to improve cost estimating methodology."

The panel actually developed twenty-four separate conclusions, which were subsequently "boiled down" to four basic recommendations; these were, again quoting from the report:

1. The panel recommends that the JLC not adopt any existing SCE model as a standard.

2. The panel recommends that a guidebook be developed that can be used by program offices to orderly qualify models and methodologies, to develop better software cost estimates throughout the entire software life cycle.

3. The panel recommends that JLC sponsor a program to implement an improved SCE methodology.

4. The panel recommends that JLC appoint an existing Government agency as an SCE data base repository and empower this agency to develop data collection standards.

In 1983, planning began for the Orlando I conference. Again, the subject of cost estimation was considered for a panel, but the approach was somewhat indirect. For this conference the 1980 EIA forecast on computers and software was the basic issue upon which the charter of Panel C, "Cost of Ownership" was established.

In the Panel C report, as excerpted, the following recommendation was made:

ISSUE: Cost Data Collection, Cost Accounting and the Use of Predictive Models for Software Costs

Much controversy is associated with the validity of the EIA prediction of $34B to be spent on software support in 1990. In order to make any realistic projections, one must have more formalized and routine cost tracking and predicting mechanisms. The suggestion here is that costs be tracked by system (hardware and items), both with respect to efforts relating to software error correction, as well as software enhancements/modifications. These software efforts should be tracked by program element.

Additionally, the current literature should be searched to determine how much work has been done to date and published in such journals as the IEEE COMPUTER MAGAZINE, THE JOURNAL OF THE ACM, NTIS, university theses, and other commonly available sources.

This issue is also directly related to the issue of proper funding policies within the Services. Additional work should be sponsored, either to be done in-house (by one of the Services or by OSD) or contractually, to develop/adapt analytical predictive models for estimating software costs as a function of system complexity, lines of code, life cycle phase of end item, etc.

It is only through accurate data collection and extensive use of these analytical models will we ever get a thorough understanding of the future software support cost within the DOD.

It is only through accurate data collection and extensive use of these analytical models will we ever get a thorough understanding of the future software support cost within the DOD.

No specific JLC initiatives are directly traceable to the recommendations made from Monterey II or Orlando I. However, since then, significant progress has been made in the use of cost estimation models and in the data collection area.

SCOPE.

The central theme of Orlando II was "Solving the PDSS Challenge." The workshop addressed various aspects of PDSS to identify areas which offer significant payoffs in terms of cost reduction, improved system reliability, streamlining of the PDSS process, and most importantly to Panel II, effective planning and management. The challenge for Panel II, as discussed in the Orlando II Master Plan, was described as:

Successful planning for transition of new systems into operational use requires proper tools to forecast resource requirements. Accurate forecasting requires an in-depth understanding of the system design, the selected support concept, interoperability issues, system support technologies, equipment, tools, and quantities and skills of personnel. Techniques must be developed to permit proper forecasting and budgeting for PDSS activities.

ASSUMPTIONS.

In planning the approach of Panel II, the following basic assumptions were made:

- The model(s) selected will be applicable over the entire software life cycle; PDSS is after all, as described in the report from Orlando I, a mini-life cycle type development.

- The model should be applicable to multiple software development and maintenance user communities, e.g., PDSS centers, IV&V, program management, the cost accounting community, the system analysis community, etc.

- The model should be integrated with, or be part of, the overall PDSS software support environment (SSE).

- Although the model will emphasize software support, its boundary limits should be defined to complement, but not duplicate and overlap, predictions better obtained from hardware forecasting models, e.g., handle software related products and effort not covered in hardware cost estimation or analysis models.

APPROACH.

First Day (Monday).

After the opening Workshop Joint Session on Monday afternoon, the eight workshop panels met in their assigned rooms for the first time.

The purpose of the initial Group Session was:

1. To review the panel's objectives and purpose.

2. To stress the requirements to provide objective, well-defined recommendations to the JLC JPCG-CRM on the issue of Forecasting PDSS Resource Requirements.

3. To discuss the general approach to the panel's operation, schedule, administrative detail, room locations, etc.

4. To discuss the planned approach to Subpanel (breakout groups), Group, and Joint Sessions.

5. To allow the co-chairs and each member of the panel to introduce herself/himself to the group.

Monday and Tuesday Group Sessions.

Selected members of the panel were contacted prior to the workshop and requested to present briefings (on Monday afternoon and Tuesday) on selected forecasting models currently utilized by the Services and industry. Briefings were from 30-45 minutes in length or longer, as required. Briefing timing and schedule were coordinated with and by the co-chairs. The briefings were intended to provide insight and responses to selected issues addressed by the panel.

Briefings on forecasting models which were presented are identified in the List of Briefings.

Wednesday and Thursday Subpanel Sessions.

The panel broke out into subpanels to discuss the following topics:

1. Management Issues - Related to the selection, use control, and qualification of selected models.

2. Technical Issues - Related to the application, tailoring, and calibration of selected models.

Wherever possible, members were allowed to select the panel/topic area of their choice.

Subpanels discussed their assigned topic areas, identified planned recommendations in accordance with the panel outline, and prepared written notes on major items of discussion/decisions. It was found that a significant amount of overlap in subject matter occurred in the two subpanels as related to five basic issues: PDSS Forecasting Problems, Standard Forecast Models, Model Characteristics, Model Criteria and Future R&D requirements. New subpanels were formed on Thursday to merge the management and technical viewpoints under these five basic issue topics, and to evolve the panel's basic recommendations.

Issues of a general nature, or those which may have the potential for impacting another subpanel and/or panel, were identified by subpanel leaders and reported to one, or both, panel co-chairs. The co-chairs facilitated/coordinated required panel/subpanel interaction.

Thursday Afternoon Workshop Group Session.

The subpanels reformed for a Group Session in the afternoon on Thursday. Subpanel leaders and/or recorders provided a review of the subpanels' deliberations, and reported on the recommendations developed by the subpanel.

The co-chairs prepared, based on these subpanel reports, a panel summary for presentation at the following Joint Session.

Final Workshop Joint Session - Friday.

For the final workshop Joint Session, each panel prepared and presented a 20-30 minute briefing. The briefing summarized the panel's work during the week, provide a review of the panel's recommendations to the JLC JPCG-CRM and provide other germane and salient information.

Preliminary Written Report - Panel co-chairs and participants prepared and submitted to the JLC Workshop Committee, a written, preliminary report based upon draft written material, prepared at the workshop and on the final Joint Session briefing.

Final Written Report - Panel co-chairs, subpanel leaders and recorders have developed, prepared, reviewed, coordinated and issued this final panel report to the JLC PDSS Workshop Committee in accordance with the JLC's and the panel's schedule. The report presents details on the panel's charter, deliberations, and recommendations.

DISCUSSION AND RECOMMENDATIONS.

Successful planning for the transition of new or modified systems into operational use requires proper tools to forecast resource requirements. Techniques which provide high levels of management, confidence, and support must be developed to permit accurate forecasting and budgeting for PDSS activities.

Panel II identified the following basic problems in the forecasting of PDSS resource requirements.

1. Currently the estimation of PDSS resource requirements is largely unstructured and nonstandard when viewed across the Services.

2. There is not a designated Service level authority responsible for establishing guidelines for PDSS resource forecasting methodology.

3. Current forecasting techniques are not based on a valid historical data base for each PDSS center.

4. There is not a common definition of software development and PDSS terms or activities across DOD organizations.

5. There is a lack of objectivity in current estimating techniques.

6. Current techniques are often used to "back-in" to a pre-established, or approved budget, rather than to establish the actual required budget.

7. Those using and/or inputting data for a forecasting technique are not adequately trained.

8. The lack of a historical data base makes it difficult to predict change rates and resulting PDSS resource requirements during the development and support processes.

9. The lack of a current, validated historical data base causes forecasting techniques to have limited acceptance by management.

10. There are limited means for high level management to assess the impact of changes in funding levels, personnel allocations, or government/contractor support ratios on the acquisition and support of software.

The following recommendations are made by Panel II to provide the JLC JPCG-CRM with a course to follow, which will lead to a more effective method of forecasting PDSS resource requirements.

RECOMMENDATION 1 (Recommendation 4-2-01) - ESTABLISHED SERVICE METHOD.

Near Term: The JLC JPCG-CRM should support the establishment, on a Service basis, of a policy and implementing mechanism which directs a COCOMO - like method to be used for forecasting software development and software support resources.

From panel discussions, it was found that all of the Services were predominantly applying some extensions of COCOMO. To date, the Army Life Cycle Software Engineering community has adopted a COCOMO-based model called the Software Engineering Cost Model (SECOMO) as its standard for software resource forecasting. The Marine Corps is in the process of gaining acceptance for their COCOMO-based model as its standard for the forecasting of required software maintenance resources. The Air Force and Navy have not adopted a standard SCE model, but have used COCOMO techniques for some of their software forecasting.

COCOMO's use as a de facto Service SCE model is in part attributable to its nonproprietary status. Its use is not restricted, due to software data rights concerns. This, in turn, permits tailoring and common usage of the method by industry and government with minimal restrictions and cost.

The immediate establishment of a policy and implementing mechanism, which directs that each Service utilize a COCOMO-like method, will help to quickly formulate a standard technique for forecasting PDSS resources.

The pertinent characteristics desired in a standard SCE forecasting model are as follows:

1. The model must address activities and resources in a PDSS environment.

2. The standard PDSS forecasting model should conform to DOD-STD-2167 and other related DOD standards.

3. The model should support detailed cost, manpower, and schedule forecasting over the full life cycle.

4. The model should be accurate, easily understood and accepted by management.

5. The model should be adaptable to unique Service requirements.

6. The model should have operational usage characteristics which are easy to use, portable, interactive, and contain easy-to-read output.

7. The model should be well defined and supported by documentation, training, and Service implementation policy.

8. The model should be flexible and extendable to allow incorporation of changes based on continuing research.

9. The model's operational cost should be reasonable so that frequent reuse is not prohibitive.

RECOMMENDATION 2 (Recommendation 4-2-02) - STANDARD DATA BASE.

Near Term: The JLC JPCG-CRM should sponsor an initiative to establish, on a triservice basis, a standard software data collection initiative and a supportive standard data definition initiative. Although the basic methodology structuring COCOMO is sound, obtainable results today will at best be a "ballpark" estimate, since modeled computational variables are based on multiapplication, industry data collected in the 1970s. Through application of specific software data collection, models can be statistically calibrated to more accurately predict costs, schedule and other resource requirements. This, in turn, promotes more confidence in obtainable results. Presently, there are no common data definitions of software development and PDSS terms and activities across DOD Services. By standardizing on a SCE technique, standard data definitions will be more easily formulated. Standard data definitions development is needed to establish data collection criteria. Also, a prescribed Work Breakdown Structure for software data elements compatible with MIL-STD-881 Revision A (1 Dec 86) and DOD-STD-2167 must be defined to promote consistency for all data collection among systems. Data definition and collection initiatives on a triservice basis can produce the broadest maximum consistency for collecting software data from developing contractors, support contractors and in-house, government support.

RECOMMENDATION 3 (Recommendation 4-2-03) - SCE TRAINING.

Near Term: The JLC JPCG-CRM should encourage the Services to define and implement a management and technically based training program to support the effective use, analysis, understanding and acceptance of SCE method(s).

As with any new technology, SCE model training for nontechnical support personnel, technical personnel and management is required. Without adequate training, nontechnical model users have difficulty understanding and implementing the model, technical personnel have trouble inputting appropriate data, and management does not know the basis or the accuracy of results provided to them. Technically based training should help to minimize the "garbage-in and garbage-out" syndrome that results

in a loss of confidence and credibility in modeled results; even where model algorithms may be accurate. Management based training promotes understanding and confidence necessary for management acceptance.

RECOMMENDATION 4 (Recommendation 4-2-04) - FUTURE REQUIREMENTS.

Near to Mid Term: The JLC JPCG-CRM, through the SEI and STARS JPO, should provide leadership toward the establishment of a Service-oriented research program to develop and promote the insertion of new and evolving technology in SCE methodologies.

Off-the-shelf models such as COCOMO, while well defined in limited areas, do not address all software resource forecasting needs for each Service. Further research and investigation is needed in areas that expand existing SCE model capabilities, integrate the software model in the life cycle process, and determine resource forecasting needs that support merging software technologies. For long term research, DOD should establish a central authority to support the upgrading of SCE methodology to reflect emerging software technology.

The pertinent areas desired for research and investigation to expand model capabilities are as follows:

1. Tailor the SCE model capabilities to cover the software support organization's environment.

2. Ensure that the model supports sensitivity analysis, "what if" analysis, estimation of confidence ranges, and identification of high risk approaches.

3. Expand model coverage to estimate additional life cycle resource requirements such as prototyping, and requirements definitions; acquisition management; PDSS preparation; PDSS administration; facility management; contract management; system integration, test and evaluation; conversion; installation; training; data base administration and computer resource requirements.

4. Expand model coverage to complex software situations such as incremental development; multiple versions; large, loosely coupled software complexes (combinations of operational, on-line support, and off-line support software); and mixtures of government supported and commercially supported software.

5. Develop better methods for estimating the amount of software to be developed or modified.

6. Incorporate Ada language design methodologies.

7. Artificial intelligence and knowledge based systems characterization in the development and maintenance process need to be added to the SCE model.

8. The development and maintenance of embedded control systems with software using integrated circuit technology (e.g., Very High Speed Integrated Circuits) needs to be added to the SCE model. This includes appropriate characterization of the new types of hardware employed to develop and operate software such as parallel processors and distributed networks, and to incorporate new technologies such as reusable code repositories.

RECOMMENDATION 5 (Recommendation 4-2-05) - STANDARD DOD METHODOLOGY.

Long Term: The JLC JPCG-CRM's long-term goal should be to support the adoption of a standard DOD SCE model.

Without the focus created by a long-term goal of adopting a DOD standard model, each Service is likely to establish diverging COCOMO-like methods for their use. The convergence of COCOMO-like methods for SCE models can stem from each Service sharing their modeling requirements, methods and tools, to help improve approaches for estimating their software resources. A DOD standard SCE model also helps to channel creative efforts into more productive areas by filling model voids. Costs are saved by minimizing duplication of effort, while limiting the use of a second model for only independent perspective auditor review. The adoption of a DOD standard SCE model promotes consistency amongst the Services for documentation, data collection, comparison of costs, training, and decision making.

The pertinent characteristics desired for a standard forecasting model still apply. Although there is much commonality amongst the Services, the creation of one standard DOD SCE model will require flexibility to cover options that may be specific to a Service. As new standard methodologies evolve from research and investigations, their placement into one single model is only advised if it does not make the model too cumbersome and complex. If it does, another standard model should evolve to complement the requirements not covered in the one model and form one standard set of DOD models that handles all software requirements without any duplications.

PANEL II LIST OF BRIEFINGS.

1. "PDSS Resource Forecasting Model Used at Sacramento ALC," by Tom Westaway, SM-ALC/MMARA, (916) 643-6388.

2. "SECOMO: The Army's Implementation of COCOMO," by W.B. Goethert, ITT Research Institute, (315) 336-2359.

3. "SECOMO Improvements," by Milt Winter, CECOM, (201) 532-1222.

4. "Software Cost Estimating of Command, Control, Communications and Intelligence Systems," by Jack Sterling, CECOM, (201) 544-4119.

5. "Ada COCOMO Overview," by Barry Boehm, TRW, (213) 535-2184.

6. "Selection of Investment Strategies Which are Optimal for the Implementation of Software Support Environments," by Ken Nidiffer, Software Productivity Consortium, (703) 391-1820.

7. "Use of COCOMO on the STARS-SEE," by H.G. Stuebing, NADC, (215) 441-2314.

8. "Software Quality Approach: Reliability, Availability, and Maintainability," by Elmer Branyan, General Electric, (215) 531-1001.

9. "USMC Software Support Personnel Requirements Model," by LtCol R.L. Pollard, Jr., MCDEC, (703) 640-2546.

10. "NUSC Model Applications," by Ron Leask, NUSC, (203) 440-4366.

11. "Software Support Model for Reliability Prediction," by Dennis Wood, Software Enterprises Corp., (818) 889-7814.

(Intentionally Blank)

PANEL II BIBLIOGRAPHY.

1. DOD-STD-2167, "Defense System Software Development," 4 June 1986

2. DOD-STD-2168, "Defense System Software Quality Program," (Draft) 1 Aug 1986

3. JLC, "Proceedings of the JLC JPCG-CRM on Computer Resource Management, Second Software Workshop (Monterey II)," 1 November 1981

4. JLC, "Final Report of the JLC Workshop on PDSS for MCCS (Orlando I)," Vol. I - Executive Summary, and Vol. II - Workshop Proceedings, June 1986

5. DACS, "STARS Measurement Survey Summary," May 1986

6. RADC-TR-84-156, "Cost Estimation Techniques for C3I System Software," July 1984

7. U.S. Army AMC, "MS-3 Model Evaluation of Constructive Cost Model (COCOMO)," June 1985

8. STARS Business Practices Area Management Workshop, Panel 2 Program Estimating and Scheduling -- Briefing Papers and Final Report, November 1985

9. STARS, "Functional Task Area Strategy for Measurement," 30 March 1983

10. NSIA, "Software Quality and Productivity Conference Proceedings," 12-13 March 1986

(Intentionally Blank)

SOFTWARE CHANGE PROCESS
PANEL III
PROCEEDINGS

Due to the broad issues needing to be addressed pertaining to the Software Change Process topic, Panel III was divided into two separate panels. Their reports follow.

Panel IIIA - PDSS Modeling/Support Strategies.

Panel IIIB - Configuration Management.

PANEL IIIA - PDSS MODELING/SUPPORT STRATEGIES

OBJECTIVES.

The objectives of Panel IIIA were to:

1. Identify the functions involved in the software support process, model that process, and develop a contingency model.

2. Identify software support strategy alternatives.

BACKGROUND.

DOD-STD-2167, which describes the software development process, has received broad review and acceptance from industry and each of the Services. The challenge that remains is to create a standard which describes the PDSS process and is consistent with the framework provided by DOD-STD-2167 for software development.

Orlando I provided a model of the software development process; however, it did not adequately address PDSS. This model, the Orlando II model, uses the software development process contained in DOD-STD-2167 as a starting point and expands it to include the unique PDSS activities not included in the Orlando I model.

SCOPE.

The software support process is part of a system support process. Panel IIIA did not consider those activities that are "system support activities." System support activities include, for example, hardware and software integration. Therefore, the software support process model includes only those activities that comprise the software support process and not part of a larger process.

ASSUMPTIONS AND CONSTRAINTS.

1. The software support process is initiated by the receipt of a system change request. We assumed that the classification of change requests into hardware or software is a system support activity.

2. We assumed that the PDSS activities conducted during initial software development have been satisfactorily completed.

3. The model assumes that the software being supported has been baselined.

APPROACH.

The panel was divided into three teams. An objective or interim objective (a logical progression towards the objective under consideration) was presented. The objective was discussed and assumption and boundaries were established when appropriate. Each team withdrew into a team session for a predetermined period to develop a team solution. The teams rejoined and each team solution was presented by the team captain. Team captain responsibilities were rotated among each team member. A question period was allowed after each team's solution was presented. During this period only team captains were permitted to talk. After all three solutions had been presented, a general session was conducted, with all members participating, to analyze the solutions and consider dropping a solution from consideration or to develop a hybrid solution by combining elements of two or more team solutions. The team captains polled team members to determine the team vote. The Panel chairman polled the team captains to determine the Panel solution. A simple majority of the three teams constituted a winning vote. This methodology, although not efficient, did encourage participation by each member; provided for an equitable decision making process which reflected the consensus of the membership; and precluded domination of minority opinions. It should be noted that we were forced to abandon this methodology about half way through the week or risk not completing our panel objectives.

POST DEPLOYMENT SOFTWARE SUPPORT MODEL.

PDSS. Realizing that DOD has still not adopted a definition for PDSS, our initial task was to agree on a definition. Although many alternatives were considered, the panel concluded that the definition of PDSS recommended by Orlando I remains correct and applicable. The Panel recommends that the Orlando I definition be adopted and implemented by DOD. For convenience, the Orlando I definition of PDSS is provided:

"Post Deployment Software Support (PDSS) is the sum of all activities required to ensure that, during the production/ deployment phase of a mission critical computer system, the implemented and fielded software/system continues to support its original operational mission, subsequent mission modifications, and product improvement efforts."

PDSS Flow Overview. Figure 2 depicts the Orlando II PDSS process model. The PDSS process consists of three phases: Phase I (initial analysis), Phase II (software development), and Phase III (product logistics). The output of one phase provides the input to the next. Phase II is the software development model contained in DOD-STD-2167. Phases I and III, which include mostly management and support activities, are new distinctions.

The final model (Figure 2) is simpler than the Orlando I model, clearly identifies the activities that occur in the PDSS process, and provides a logical and distinct separation between each phase. The last consideration is important because Phase II is frequently contracted, while Phases I and III are most often performed by the Services. Additionally, the model incorporates logistics activities which are not incorporated in the Orlando I model or in DOD-STD-2167.

The software change request (SCR) initiates the process. These can take the form of problem reports carried over from initial development or submitted by operating forces during system deployment. SCR's also include changes due to newly defined user requirements or to changes in the operating environment.

The first phase (initial analysis), is a set of specialized activities which transform software change requests into an approved Engineering Change Proposal (ECP). The release of the ECP to the next phase is usually the prerogative of the system program manager/sponsor.

The second phase (software development), is described in DOD-STD-2167. It involves the design, development, integration testing and operational testing of a new baseline software configuration. This process transforms a software change ECP into an approved release package which is a new baseline change to the operational baseline in use by operating forces. The release of this new baseline is usually approved by the system PM.

The third phase (product logistics), implements the decision to deliver the approved release package to the operating units. This can include training activities, production changes during manufacturing processes, documentation publication and distribution, and installation of the program on deployed systems.

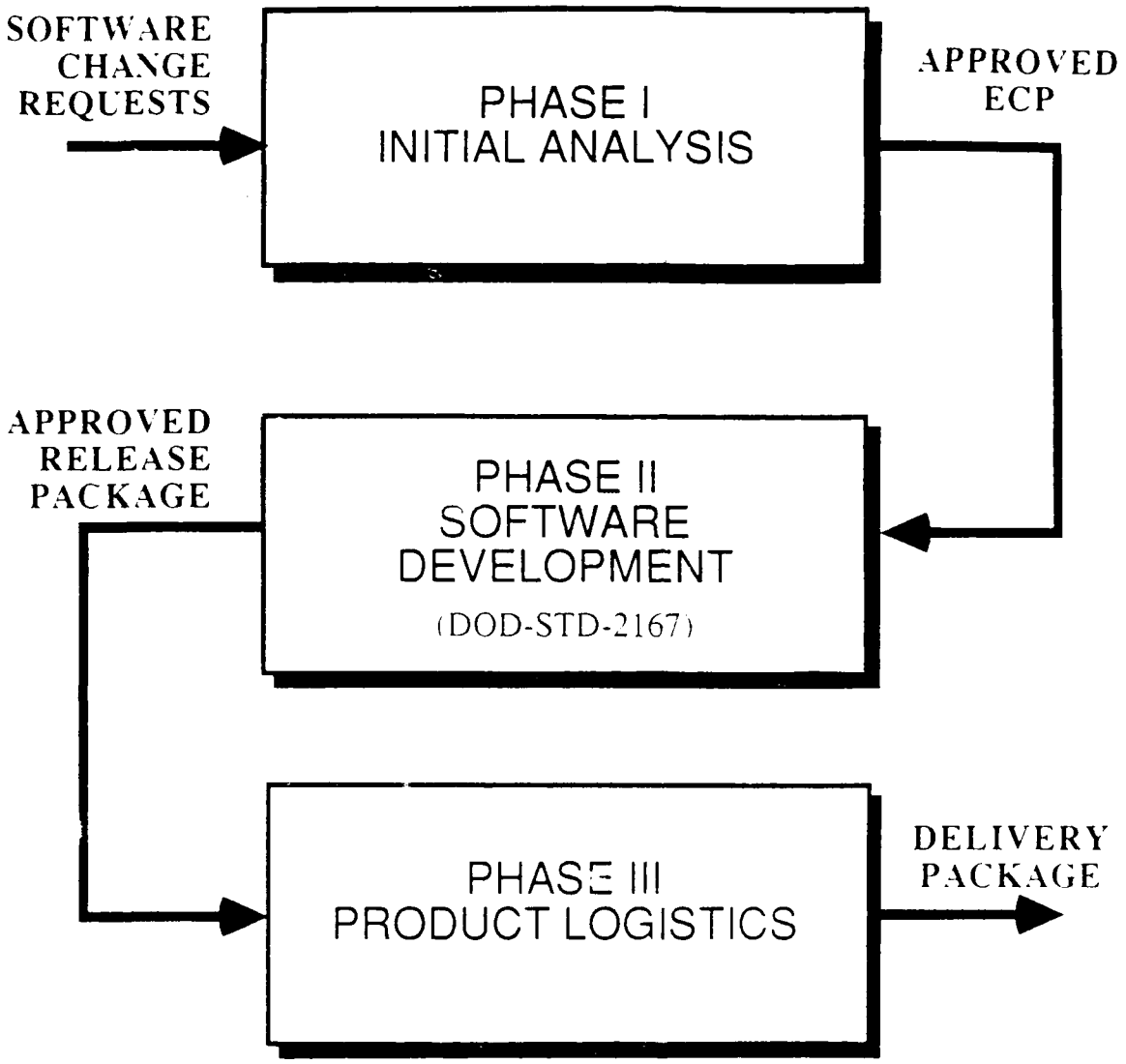


FIGURE 2. PDSS Process

PDSS DETAILED MODEL.

In order to develop an aggregate model which was both complete and understandable a simplified set of distinctions had to be developed to describe the functional categories which comprise the PDSS process. These are management, technical, and support. Management includes configuration management, coordination, control, resource allocation, and decision making activities. It is understood that there is management involvement at all levels, but only significant management controls and decision points are shown in the model. Technical functions are those involved with software engineering and testing. Support resources include hardware support, logistics/supply support, and general overhead activities. In general, the management activities described in the model are not contractible, whereas many of the activities in the other two functional categories could be contracted by the government. The PDSS detailed model is presented in Figure 3.

Initial Analysis. During the initial analysis phase, software change requests are transformed into an ECP. The term software change request is used in a generic sense to include recommended changes to correct latent errors (i.e., to ensure the software supports its original operational mission); to accommodate changes in the environment (i.e., to ensure the software supports subsequent mission modifications); and product improvements (i.e., product improvement efforts). Software change requests are received and tracked to provide status accounting of all perceived problems associated with the software.

A Software Configuration Control Board (SCCB) is convened by the PDSS Activity (PDSSA). At this session the resources are allocated to perform an initial analysis. The initial analysis is primarily a technical activity whose purpose is to recreate the fault based on the input provided, isolate the source (hardware versus software), attempt to isolate the error in the software in order to estimate the level of effort involved to correct the problem. Also internal and external coordination is undertaken to identify interoperability issues associated with a particular change. The output of this effort is an approved analysis package.

Support analysis is performed upon the approved analysis package. The result of this activity is the validation of the change, identification of the logistical impact of the change, and identification of the specific resources necessary to effect the change.

The analysis effort provides the necessary information for a second SCCB review. At this review recommended changes are approved for implementation (implementation decision) and prioritized. It should be noted that class I changes would

typically be forwarded to a system level configuration control board for the implementation and the priority decisions. Class I changes will normally be forwarded as an ECP.

Software Development. The process in Phase II is identical to that described in DOD-STD-2167. It involves further analysis of the ECP, design, code, unit/software integration test, system test and operational test.

The set of activities (analysis, design, code, test) prior to system level testing is called software engineering. The basic condition for this effort is availability of the Software Development Environment (SDE). The SDE is that set of facilities and resources necessary to analyze the fault, create, design, configuration manage and test the software product.

This new software test version is then integrated into a hardware/software system utilizing resources of the System Integration Environment (SIE). The SIE usually includes the tactical system hardware and simulators/emulators necessary to drive the tactical system. This environment is designed to verify performance at the system functional specification level through testing of the system in a simulated environment. Interoperability requirements should be tested within the SIE.

The system, to include the new software version, undergoes an operational test (OT) in operational conditions. This test serves the same purpose as the OT conducted prior to the production decision during the acquisition cycle. It should be noted that this activity, because of its associated expense, is normally not accomplished for less than major systems.

The output of this process is an approved release package. This package represents all necessary software, documentation, training requirements, manufacturing (hardware) engineering change proposals, and logistical requirements necessary to effect the change.

The approval of a release package for implementation is normally made at a level higher than the system program manager.

Product Logistics. The purpose of the product logistics phase is to transform the approved release package into software and training requirements or programs for operational forces. In this effort software is generated for each weapon hardware variant. Training of the operational forces in the new software baseline is undertaken. Copies of software and documentation are reproduced and delivered. Software installation for each deployed weapon system is performed and verified. Hardware production lines are transitioned to the new hardware product baseline. Hardware is delivered and configuration managed.

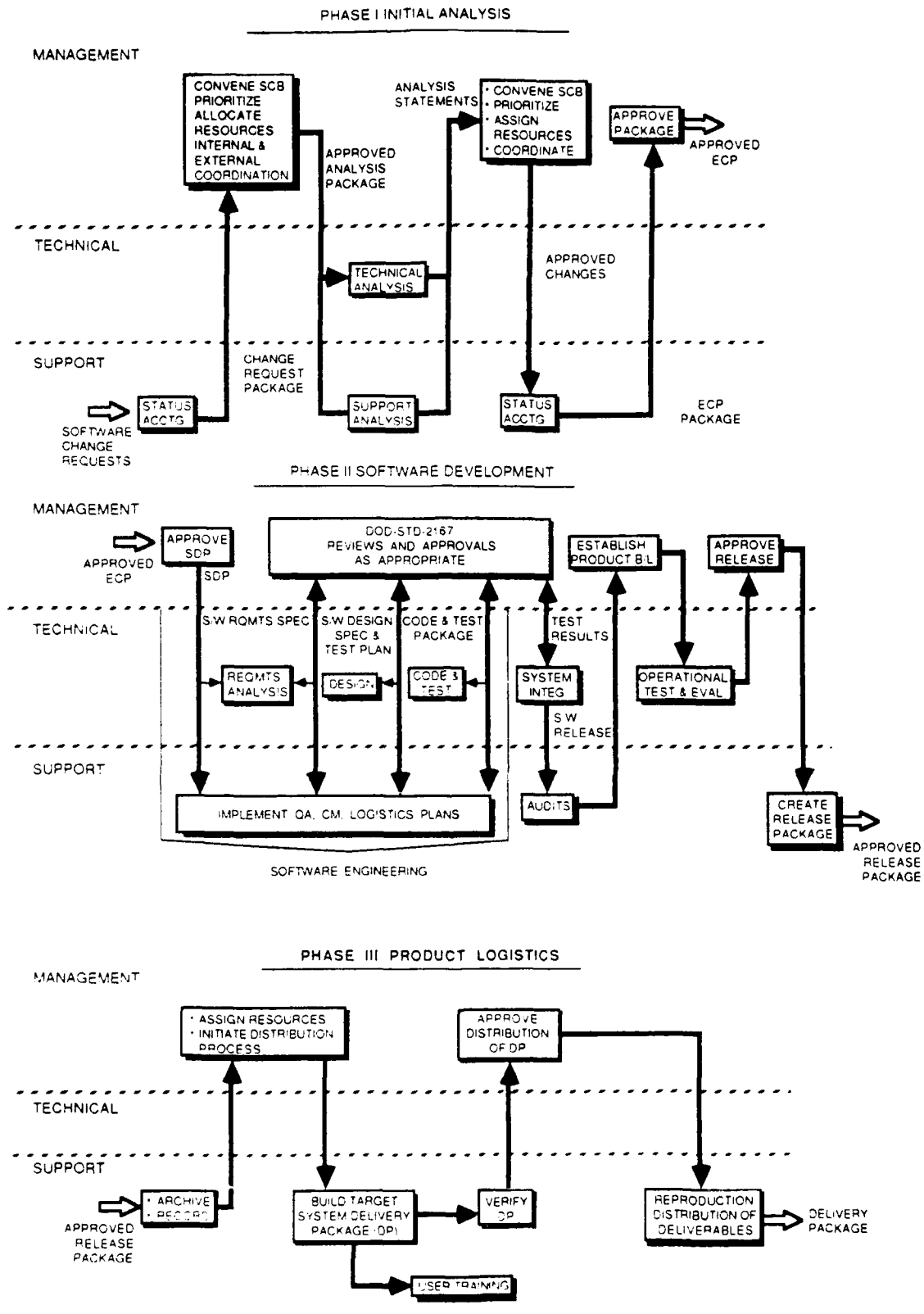


FIGURE 3. PDSS Detailed Model

PDSS CONTINGENCY MODEL.

The panel concluded that the contingency model is identical to the basic PDSS model (Phases I, II and III). However, for contingencies, management should compress the time and effort expended to complete each activity in order to satisfy the requirements of the contingency. It is anticipated that quick reaction changes will be rare occurrences and normally associated with life threatening situations. Technical documentation and all but essential training may be limited. Full documentation and training should occur as part of the next update cycle.

STRATEGY ALTERNATIVES.

The post deployment software strategies were developed based on several assumptions and observations as to the high cost factors for PDSS. The first of these is that the government can neither afford to provide full PDSS support for all Tactical Data Systems (TDS's) software nor is that level of support necessary for all TDS's software. Certain functions of the PDSS process can be contracted. The SDE and SIE are high cost items for PDSS. The ownership of the SIE and SDE is a primary consideration in arriving at a PDSS strategy decision.

Selection of Software to be Supported. In the F-14A fighter there are three digital processors. In the updated F-14D there are over thirty processors. The proliferation of digital technology in modern weapons requires discrimination as to which processors require what level of PDSS.

The three levels of software change activity are considered to distinguish the appropriate level of software support are:

a. Frequent Change. This includes software systems which require frequent change due to the nature and criticality of their function. This category would include software which exists in a volatile operational environment; frequent software changes to satisfy user requirements; and software that is accepted from the developing contractor with a significant number of latent errors which must be corrected during the deplcyment phase.

b. Occasional change. This would include systems like microcode in processors and specialized processing software.

c. Infrequent change. This would include systems used as control functions for hardware subcomponents such as engine controllers and processors in auxiliary equipment.

Selection of Ownership of SIE and SDE Facilities. Whether the government or contractor owns the SIE and SDE, or whether these facilities exist at all, is of significant importance in

determining the flexibility of the government in performing PDSS. If the government does not own these facilities then it is driven to a sole source relationship with the owner of these facilities in order to perform PDSS.

Strategy Options. PDSS activities which can be accomplished by contracting Services are in the technical and support area. Two significant cost drivers are the software engineering and system integration functions in Phase II (software development) of the PDSS model. The strategy options listed below are based on the government's ability to perform either the software engineering or the system integration/test functions.

SOFTWARE ENGINEERING FUNCTION
Government does software engineering in-house
Government sole sources software engineering
Government competes software engineering

SOFTWARE INTEGRATION/TEST FUNCTIONS
Government does integration test in-house
Government sole sources integration test
Government competes integration test

If the government owns both SDE and SIE then any combination of the software engineering alternatives and software integration/test alternatives may be combined to form the most advantageous support strategy decision. This situation, government ownership of both environments, offers the most flexibility and broadest range of government alternatives.

Ownership of these environments may be vested in the government, may be divided between industry and government, or may be totally vested in industry. On the other hand there is the possibility that either of the environments may not exist at all.

The combination of software volatility and environment ownership are determinants for limiting the PDSS strategy decision. Alternatives range from the most restrictive PDSS situation of having no capability to provide PDSS (i.e., neither SDE or SIE are owned by the government or the developing contractor) to the most flexible situation of government ownership of both SDE and SIE. The possible alternatives are shown in Table 1.

Table 1 represents the PDSS support alternatives available to the government based on SDE/SIE ownership and the level of software change anticipated. Under the circumstances certain alternatives are not recommended. For example, it is not recommended that the government procure SDE or SIE facilities to support processors where occasional or infrequent software change is anticipated. Also, the obvious case, it is recommended that the government not allow a situation where the SDE or the SIE do not exist for software that will require even infrequent change.

TABLE 1. Alternative PDSS Strategies

S/W VOLATILITY / GOVERNMENT OWNS	FREQUENT CHANGE	OCCASIONAL CHANGE	INFREQUENT CHANGE
SIE & SDE	IA, IC, IIA, IIC	NOT RECOMMENDED	NOT RECOMMENDED
SIE ONLY	IB, IIA, IIC	IB,IIA, IIC	NOT RECOMMENDED
SDE ONLY	IA, IC, IIB	IA,IC,IIB	NOT RECOMMENDED
NEITHER	IB, IIB	IB,IIB	TREAT AS HARDWARE
NONE EXISTS	NOT RECOMMENDED	TREAT AS HARDWARE	TREAT AS HARDWARE

STRATEGY OPTIONS:

- IA GOVERNMENT DOES SOFTWARE ENGINEERING
- IB GOVERNMENT MUST DEPEND ON DEVELOPING CONTRACTOR FOR SOFTWARE ENGINEERING
- IC COMPETE SOFTWARE ENGINEERING
- IIA GOVERNMENT DOES INTEGRATION/TEST
- IIB GOVERNMENT MUST DEPEND ON DEVELOPING CONTRACTOR FOR INTEGRATION/TEST
- IIC COMPETE INTEGRATION/TEST

The facilities investment decision must be made early in the development phase and always prior to the full scale development (FSD) contract. These considerations should be performed as part of a Software Support Requirements Analysis (SSRA) and documented in the CRLCMP.

CONCLUSIONS.

Impact if PDSS Model not Implemented. Without a clear understanding of the PDSS process and the included activities, the PDSS process will be difficult to manage or standardize at the DOD level. The PDSS model, presented in Figure 3, includes the management, technical and support activities that must occur during PDSS. Since those activities in Phases I and III are not considered in DOD-STD-2167, the DOD-STD-2167 model is not complete when applied to PDSS.

The absence of clearly defined government PDSS management responsibilities and corresponding controls often results in an ineffective and costly software change process. Failure to analyze the requirement for government ownership of the SDE/SIE often result in a de facto selection of PDSS strategy. Failure to plan for the procurement of the SDE/SIE often results in limiting otherwise available support alternatives. The ownership of a SDE or a SIE is such a critical decision, that it must be documented in the CRLCMP and reflected in contractual documents for FSD. Otherwise, PDSS requirements, in support of the PDSS strategy decision, will not be satisfied.

ANTICIPATED BENEFITS.

A standard definition will promote common understanding of the PDSS process and the activities involved. For example, all three categories of software change will be considered by all as part of PDSS. This may have significant funding consequences and should be standardized within DOD.

It is difficult to standardize a process without first describing it or modeling it in some manner. The completed PDSS model will allow the DOD to establish process standards. The approach used in the model clearly demonstrates the relationship between the initial software development process and the software support process.

MANAGEMENT CONTROL.

Government management controls and responsibilities of the PDSS agency should be clearly defined regardless of the support strategy selected.

PLANNING.

The affect of software volatility during the deployment phase and software criticality are important considerations to be analyzed when considering the PDSS strategy. Additionally, the ownership of the SDE and the integration/test environment determines the support options that are available to the government. Therefore, these determinants must be the result of analysis and conscious decision by the government early in the software development phase. The decisions and rationale for these decisions should be reflected in the CRLCMP and become the basis for the definition of PDSS requirements.

RECOMMENDATIONS.

a. Recommendation 4-3-01. That the definition of PDSS developed at Orlando I be approved as a DOD standard and implemented in appropriate regulations.

b. Recommendation 4-3-02. That the software support model described in Figures 2 and 3 be further developed and refined. A standard software support process model, based on the approach presented herein, should be adopted by DOD.

c. Recommendation 4-3-03. A SSRA should be performed to determine the PDSS requirements.

d. Recommendation 4-3-04. Management of the PDSS process must be vested in the government. Therefore all planning documents such as the CRLCMP, the Test and Evaluation Master Plan (TEMP), the Configuration Management Plan (CMP), the Quality Assurance Plan (QAP), and the Software Development Plan (SDP) must specifically address management controls to be taken by the government.

e. Recommendation 4-3-05. Resources required for the SIE technical and SDE should be obtained based on an approved PDSS strategy.

f. Recommendation 4-3-06. A PDSS strategy must be established in the CRLCMP by analyzing the volatility of the software product, and cost of ownership of the SDE and the SIE.

PANEL IIIB - CONFIGURATION MANAGEMENT

OBJECTIVES.

CONFIGURATION MANAGEMENT POLICY.

Identify software and firmware related deficiencies in DOD configuration management directives and standards as they pertain to PDSS, and develop a recommended approach for implementing required changes.

CONFIGURATION STATUS ACCOUNTING.

Develop basic procurement documents for the development of an automated standard software configuration status accounting system.

BACKGROUND.

CONFIGURATION MANAGEMENT POLICY.

DOD Configuration Management Infrastructure. Configuration management (CM) is one of several major acquisition disciplines selected by the Secretary of Defense for management under its standardization program. The Under Secretary of Defense for Research and Engineering is responsible for prescribing overall management policy for DOD CM practices. Following the acquisition and deployment of configuration items, the Assistant Secretary of Defense for Manpower, Reserve Affairs, and Logistics is responsible for ensuring effective implementation of approved DOD CM policies and guidelines. The DOD Configuration Management Committee (DCMC), with representatives from each of the DOD components, provides necessary support in the conduct of the DOD Configuration Management Program (DCMP). Until recently, the Navy was the designated lead DOD component for configuration management. The Secretary of the Navy had delegated this responsibility to the Chief of Naval Material, who provided the chairman for the DCMC. However, when the Naval Material Command was dissolved in early 1985, lead component responsibility for configuration management was returned to the Office of the Secretary of Defense (OSD) level.

Within OSD, the CM program was previously under the cognizance of the Defense Materiel Specifications and Standards Office (DMSSO). This office was reorganized into three new offices, namely, the Standardization Program Office (SPO), the Defense Data Management Office (DDMO), and the Defense Products Standardization Office (DPSO). The DDMO is now responsible for managing the DCMP and is presently performing lead DOD component responsibilities for CM.

Configuration Management Program Status. A major overhaul of the DCMP was initiated in 1979, with initial efforts centered on updating the July 1974 Joint DOD Services and Agency Regulation on Configuration Management. Although a new draft revision of this regulation was published in 1981, the DCMC could not reach a unanimous agreement for its formal release. Because the lower level directives and associated standards could not be updated until resolution was reached by all the Services on the Joint Regulation, the overhaul process came to a standstill. As a result, the 1974 version of this regulation is still the effective version, and the planned update of the associated standards was never initiated by the committee.

When the Naval Material Command was dissolved in 1985, cognizance of Navy CM was passed to the Chief of Naval Operations (OP-04), and lead Service responsibilities for CM were assumed by OSD. To further aggravate the situation, there has been a complete turnover of all key Service representatives on the DCMC during the past eighteen months. Currently, the key DOD CM personnel are:

Mr. Carl Berry	Defense Data Management Office Director
Ms. Linda Burgher	Defense Data Management Office
Mr. John Holovet	Army (AMC)
Mr. Emerson Cale	Navy (OPNAV)
Maj Jean Kopala	Air Force (AF AM)

DOD-STD-2167 Related Upgrade Efforts. In 1978, the JLC initiated an effort to develop a joint Service military standard on defense system software development. In developing this standard, it was determined that related changes were required to three existing military standards, namely, MIL-STD-483, MIL-STD-490, and MIL-STD-1521A. These standards, which are under the cognizance of the configuration management standardization area, were modified by the JLC to ensure compatibility with the new standard. This effort culminated in the June 1985 promulgation of the new software development standard, DOD-STD-2167, together with revised versions of the three configuration management standards, MIL-STD-483A, MIL-STD-490A, and MIL-STD-1521B.

However, these latter three standards were never formally coordinated within the DOD configuration management community as required by OSD, and were therefore only conditionally approved subject to agreement being reached by the DCMC.

Planned Configuration Management Update. The DCMC plans to overhaul the entire configuration management standardization program. A revised version of Department of Defense Directive (DODD) 5010.19, the top level DOD directive on configuration management, is currently in the chop cycle. Once this directive is approved, the committee plans to begin a formal coordinated update of all related CM directives and standards. The committee has agreed to use the recommended changes developed by the Orlando II Configuration Management Subpanel as the basis for the initial update of these documents.

SOFTWARE CONFIGURATION STATUS ACCOUNTING.

Orlando I Efforts. In June 1984, the Orlando I Configuration Management Subpanel recommended that the JLC should support the development of a common automated configuration status accounting (CSA) data base system for use by all Services during development and PDSS. The JLC JPCG-CRM decided subsequently that the issues should be studied further before making a final decision on the subpanel's recommendation. A task was initiated by the PDSS Subgroup in August 1986, to perform this study, which resulted in the publication of a formal report on 24 January 1987. The report addressed several critical areas related to CSA, including:

- Software CSA data elements.
- Centralized/distributed approaches to CSA.
- CSA report generation.
- CSA data exchange.

Orlando II Efforts. The Orlando II Configuration Management Subpanel was tasked to develop high level requirements for an automated standard CSA system. It was intended that the above report, together with other related procurement documents to be provided, would be used by the subgroup in developing these requirements, and that these requirements would form the basis for a final recommendation regarding the development of the proposed CSA system.

SCOPE.

CM Documentation Review Effort. The scope of the Configuration Management Subpanel's documentation review activities was limited primarily to identifying only those deficiencies judged to have a major adverse impact on software CM practices and procedures used during PDSS. The review report is not intended to substitute for the required formal DOD standardization review cycle, rather, it is intended to aid DOD in developing initial draft updates to

applicable CM documents as a necessary first step of the update process.

Software CSA Efforts. Because of the relatively short time frame of the workshop, the Configuration Management Subpanel limited its CSA efforts to include only the implementation independent aspects of a standard automated software CSA system. The documents produced by the subpanel are not intended to be complete and final procurement documents, rather, it is intended that they will be used as a baseline for the development of specific, implementation oriented procurement documents.

ASSUMPTIONS AND CONSTRAINTS.

CM Documentation Review Effort. Based on information provided by the PDSS Subgroup, the Configuration Management Subpanel assumed that the proposed MIL-STD-2168 would be merged with the new DOD-STD-2167A. Consequently, the subpanel did not review the draft MIL-STD-2168 nor its associated Joint Regulation.

The subpanel assumed that DOD-STD-2167/2167A would be the guiding document in the software development process. Consequently, the subpanel directed its review towards establishing maximum consistency between the reviewed documents and this standard.

Software CSA Efforts. The procurement documents developed for the proposed standard software CSA system must be usable for both new system developments and existing system upgrades, and should not require the use of any specific existing CSA system or data base management system.

APPROACH.

The Configuration Management Subpanel was divided into two subgroups, one to review configuration management documents, the other to address software CSA issues.

CM Documentation Review Effort. The first step was to identify all the DOD directives, standards, and specifications involving configuration management. This step was actually accomplished well in advance of the workshop, and each subgroup member was provided with copies of the documents they did not already have. Each member conducted a detailed review of these documents prior to the workshop, concentrating on the extent to which they addressed PDSS issues. During the workshop, members developed a matrix of the reviewed documents, and listed deficiencies by specific paragraph number together with a synopsis description of the noted deficiency. The subgroup then divided into one and two person teams, each assigned to review in more detail a specified group of deficient documents. Each team developed a detailed report on the results of this review process, including appropriate summary conclusions and recommendations. These

reports were then discussed by the entire subgroup and ultimately consolidated into the final product report.

Software CSA Efforts. The CSA Subgroup initially conducted a detailed review of approximately ten CSA related procurement documents, RFPs, formal reports, and standards that were provided to them. Additional documents brought by the members were also reviewed. Based on information obtained from these documents, the subgroup drafted generic (implementation independent) guidelines and functional specifications for use in preparing a formal SOW for the development of a software CSA system. The subgroup also developed from supplied information a comprehensive list of CSA data elements judged to be essential in any software CSA system.

PRODUCTS.

The Configuration Management Subpanel developed three products:

1. A list of recommended changes to key DOD CM policy directives and associated standards.

2. A recommendation and supporting rationale for the development of a handbook for developing and implementing software CSA systems.

3. A recommendation and supporting rationale for the development of a DOD common software CSA system.

PRODUCT #1: RECOMMENDED DOD CM DOCUMENTATION CHANGES.

Discussion. The Configuration Management Subpanel was tasked to identify software and firmware related deficiencies in DOD CM directives and standards as they relate to PDSS activities, and to develop a recommended approach for implementing required changes. The subpanel conducted a detailed review of 13 major directives, standards, and specifications dealing with DOD CM policies, practices, and procedures. The review was based on the following criteria:

1. General correctness and currency.

2. Software related CM requirements.

3. Consistency with DOD-STD-2167.

4. PDSS requirements.

Conclusions. Although the review indicated that software CM issues were addressed to some extent in the majority of the documents reviewed, they were deficient in terms of consistency with current PDSS activities and practices. This is not too

surprising since the majority of these documents were issued in the early 1970s, long before many of the current software development and support philosophies were established. The subpanel also found that the reviewed documents were generally inconsistent in their relational approach to DOD-STD-2167, which is considered to be the guiding standard for all defense system software development efforts.

COGENT FACTORS.

Solutions.

Near Term. None

Mid Term (Recommendation 4-3-07). Comprehensive update of all applicable DOD CM directives and related documents. It is recommended that:

1. The JLC request OSD (Director, DDMO) to initiate a major update of the DCMP to include the formal, coordinated, and integrated review and update of all the documents listed in the Configuration Management standardization area.

2. The JLC provide the Configuration Management Subpanel's detailed recommended changes to the DCMC, with the recommendation that they be used to establish the initial formal update baselines for the applicable documents.

3. The JLC recommend to OSD that the PDSS Subgroup be tasked and funded by OSD to conduct the formal update of the CM documents, working under the cognizance of the Director, DDMO.

Long Term. Once all the applicable documents are updated, The DCMC should ensure that the documents are reviewed periodically and are maintained up to date. The longer the period between reviews, the more difficult and costly it will be to accomplish the upgrade.

Estimated Cost. $500K

Time to Implement. 2 years

Dependencies. Agreement by OSD to do the update.

Alternatives. There are no recommended alternatives regarding the need for the update. However, OSD can elect to accomplish the update without JLC participation.

Method of Implementation. The PDSS subgroup should manage the update effort, with contractor support with oversight management provided by the DCMC.

Justification for Prioritization.

This effort is already being planned by OSD, and should be funded by OSD. There is a critical need for this update effort, as detailed below:

Anticipated Benefits.

1. Essential PDSS requirements would be properly integrated into the defense software development process, thereby providing significant life cycle cost benefits, as well as improved system accountability and maintainability.

2. DOD CM directives and associated standards would be current, and could be consistently applied to all defense software development efforts. This would significantly reduce overall DOD software acquisition, procurement, development, test, and follow-on life cycle support costs.

Impact on PDSS if not Implemented. The existing inconsistent, incompatible, and outdated CM procedures, methods, and practices will continue to be applied to DOD software development efforts. This will result in ineffective accountability and control of critical software baselines, reduced system reliability and maintainability, and paralleling increased development and life cycle costs.

DETAILED COMMENTS.

DODD 5010.19 - Configuration Management (Draft), January 1987

1. This directive should be restructured to reflect the software life cycle phases as well as the system acquisition phases, and should describe the associated activities, reviews, and products within the various phases for both hardware and software. This technique, used successfully in DOD-STD-2167 for software, has proven very beneficial for both implementation and training.

- Paragraphs 6.0, 8.0, 9.0 should reference that these activities are carried forward into PDSS.

- Clarify paragraph 7(d) to be more specific as to what point in time an event occurs. Clarify deployment -- is this delivery?

- Clarify delivery activity -- acceptance, test and demonstration, or Formal Qualification Review (FQR).

- Definitions section should be consistent in all CM directives and related standards, specifications, and handbooks.

2. Segmenting this directive into phases (or activities in FSD) would enhance the usefulness of this document. Requirements analysis, preliminary design, detailed design, etc., are as applicable to hardware as they are to software. For example, code and unit test is akin to breadboard and circuit card testing.

3. Paragraph 7 - CM is specific in the controls applied during the operational phase. Similar emphasis must be included in paragraphs 6.0 - Identification, 8.0 - Status Accounting, and 9.0 Audits, since these activities do and will prevail during this phase.

4. Paragraph 7(d) Production/Deployment - is misleading since "production" of software occurred during FSD. The only to Production/Deployment is in Figure 1 (on page 3) of DOD-STD-2167, DOD-STD-2167A (Aug 1987), and AFR 800-14 (Sep 1986). This should be clarified as to the activities, products and reviews that occur during this phase.

5. Section C (enclosure 2) - Definitions - These should be consistently applied throughout all related documents. This is especially important to those definitions contained in DOD-STD-480A and DOD-STD-2167. It should be noted that DOD-STD-480A definitions have always been considered the "official" reference for the CM function. If a universally agreed to set of definitions are to be provided then those contained in the present standards should be transferred and consolidated into the forthcoming DODI-5010.XX.

<u>DOD INSTRUCTION 5010.XX</u> - (Planned CM Implementing Instruction)

1. Include a section for the operational phase to include CM organization, identification, control, accounting and audits.

2. The format of the instruction should be the same as DOD-STD-2167 for hardware and software.

3. Incorporate appropriate sections of the Joint Regulation - Configuration Management (1974) into the new instruction.

4. As a minimum, the new instruction should address PDSS requirements and policy guidance. Whether this can include a similar section/subsection for hardware maintenance is not discussed here, but it should be considered by the Office of Primary Responsibility (OPR). For software, the intent and purpose of CM in PDSS must be addressed along with the CM organization requirements, and the activities relating to identification, control, status accounting and audit.

5. The role of CM in controlling interfaces should be addressed.

6. DOD-STD-1467 (AR) (Software Support Environment) should be used as a guide for the software section.

DOD-STD-2167 - Defense System Software Development

The deficiencies in the current DOD-STD-2167 approach to software life cycle support are:

1. No requirement to identify the software life cycle support concept during the software requirements analysis phase with review at Software Support Review (SSR).

2. No requirement to document the software life cycle support concept as part of the Software Development Plan (SDP) (DI-MCCR-80030).

3. No detail relating to CM activities in the concept exploration, demonstration and validation, or production deployment phases.

4. No provisions for the orderly transition of CM and support requirements between life cycle phases.

5. In order for the software life cycle support concept to be integrated into the software development process, there must be explicit requirements within DOD-STD-2167 to plan, integrate and transition software life cycle CM and support activities.

DOD-STD-2167A - Draft update to DOD-STD-2167

The requirements for software life cycle support are provided for in the software preliminary design and detail design phases. These development phases require the development of a Computer Resource Integrated Support Document (CRISD-DI-MCCR-80024), with reviews at the Preliminary Design Review (PDR) and Critical Design Review (CDR). In addition, the CM requirements are relegated to the Functional Software Design (FSD) phase with no CM activities identified in the other life cycle phases.

Modify the following paragraphs as specified.

- 4.1 Add new paragraph:

  "(g) Software Transition to Deployment."

- 5.1.1.3.a.3 Add: "and transition plan"

- 5.1.1.3.b Add new subparagraph:

  "(5) Plans for transition to support of deployed systems."

- 5.1.1.3.d Add new subparagraph:

"(7) CM transition planning to support organization."

- 5.1.1.3 Add new subparagraph:

"j. Software support concept and evaluation, including plans to transition the support activities to the PDSS activity.

(1) Operational Software and Documentation

(2) Support Software and Documentation.

(3) CM software, status, data and documentation.

(4) Contractor support to be supplied during transition."

- 5.1.1 Add new paragraph:

"5.1.1.12 In the definition and analysis of the software support concept, the specific requirements to define the concept will be documented in the SDP and shall be subject to contracting agency approval. The contractor shall analyze the System Segment Specification for software and computer requirements to define the support concept."

- 5.2.1.12 Replace with:

"The contractor shall define a preliminary version of the information to perform life cycle support for the contractually delivered software in an Operational Support Plan which provides for the integration for the requirements of the CRISD with the Operational Concept Document, provides for the LSA of software, and provides for the orderly transition of software support activities between life cycle phases."

- 5.2.1.12.a Add to paragraph:

a. The support environment, using DOD-STD-1467 as a guide, describing required....

- 5.3.2.10 Replace with:

"The contractor shall produce an updated Operation Support Plan."

o Appendix B Revise to include the details of CM activities during each life cycle phase, to reflect provisions of DOD INST-5010.19, and to provide for the orderly transition of CM between life cycle phases.

DI-MCCR-80025 Add to paragraph:

- 10.2.7 "The transition plan for the orderly transfer of the support to the government agencies."

DI-MCCR-80030 Add new paragraphs:

- 10.2.5.5 Software Transition Plan (STP). This paragraph shall be numbered 3.5.

- 10.2.5.5.1 This paragraph shall be numbered 3.5.1 and describe the organizations responsible for providing the STP. This subparagraph shall include authority/responsibility of each organization and its relationship to other organizational entities. A chart will be utilized to illustrate this structure.

- 10.2.5.5.2 STP revisions. This paragraph shall address the transition plan activities to be accomplished at each applicable review. The following shall constitute the minimum revision addressed.

  - Support issues must be addressed at System Requirements Review (SRR), System Design Review (SDR), Software Specification Review (SSR), and Functional Configuration Audit/ Physical Configuration Audit (FCA/PCA).

  - At the SRR, the support aspects of the system must be addressed including life expectation of the systems which may/will drive the solutions of hardware devices and their life expectancy for sparing.
  - At the SDR, the availability requirements of the hardware must be addressed.

- At the SSR, address change expectation of software and partition areas of expected change into designed area for post deployment ease of support.

- At the PDR, discuss the aspects and issues of supportability areas derived and how they were addressed.

- At the FCA/PCA, final review of support effort including planning for transition operation is understood by development and support.

- 10.2.5.5.3 Personnel. This subparagraph shall be numbered 3.5.4 and identify and describe the specific resources other than personnel, necessary for performance of the transition to the support activity. The description of each resource shall define the placement of responsibility for the management of this resource.

MIL-STD-483A - Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs.

Modify the following paragraphs:

- 3.1.1 Revise to indicate that the CM plan must address the current phase of the system CM activities, and the planning needed to support subsequent phase CM activities.

o Appendix I Replace with DI-E-2035B.

o Appendix II

- 20.6.1 Indicate that the support agency representative is a member of the ICWG.

o Appendix III Provide guidance as to applicability of the System/Segment Specification requirements (i.e., what is a System/Segment and when is it identified?).

o Appendix IX Revise to reflect a uniform minimum approach to software and software media identification and marking (e g. part number, stock number, and nomenclature).

- o Appendix XIV

  - 140.6.2 — Address the use of the Software Problem/Change Report (SP/CR) identified in paragraph 80.5.2 and DOD-STD-2167A (draft issue reviewed) paragraph 5.10.5.

- o Appendix XVII

  - 170.5.8 — Revise to indicate that the LRU level is often an appropriate level for effective configuration control during development.

MIL-STD-1521B - Technical Reviews and Audits for Systems, Equipments, and Computer Programs

Modify the following paragraphs:

- o Appendix E

  - 50.2.2.d — Add after "Computer Resources Integrated Support Document." the following: "This document should be reviewed to insure the resources necessary to support the software during operational deployment of the system are addressed."

- o Appendix H

  - 80.1.4 — Change the paragraph to read:

    "A final review shall be made of all operation and support documents (i.e., Computer System Operator's Manual (CSOM), Software User's Manual (SUM), Computer System Diagnostic Manual (CSDM), Software Programmer's Manual, Firmware Support Manual, CRISD) to check format, completeness, and conformance with applicable data item descriptions."

  - 80.3.2.i — Add "Computer Resources Integrated Support Document (CRISD)".

  - 80.4.10 — Add new subparagraph as follows:

    "i. The CRISD should be reviewed to insure the resources necessary to support the software during operational deployment of the system are addressed."

MIL-STD-490A - Specification Practices

Modify the following paragraphs:

- 3.1.3 Employ consistent use of HWCI and CSCI, (e.g. the B1 specification should only be applicable to either the hardware portions of the prime item or provisions for software requirements should be identified in Appendix II.

- 3.1.3.1 The development of the System/Segment Specification is the point at which CM begins for the system/segment under development. Wording should be added to require the contractor to institute internal CM of the functional characteristics.

- 3.3 Should be removed from this standard and placed in DOD-STD-480. This would place all engineering changes to configuration items in one standard.

MIL-STD-499A - Engineering Management

This standard was reviewed for CM in PDSS. Three references (paragraphs 4(l), 10.1.9, and 10.2.9) address interface design compatibility and documentation control, with the latter nonmandatory in both places. CM is not specifically mentioned, but may be included in the engineering specialty integration reference. PDSS, if covered, is under Integrated Logistics Support. Software is not covered adequately.

Modify the following paragraphs:

- 4(n) Add configuration management and quality assurance (QA).

- 4(s) Add new paragraph to reflect planning into the operational phase.

- 4(t) Add new paragraph to reflect software CM considerations.

- 5.1 Revise to reflect life cycle activities as well as CM and QA.

- 10.1.6 - Update to reflect [illegible] reviews now in MIL-STD-1521B.

  - Change "computer programs" to "software".

- Add tasks for PDSS and CM, as well as the combination of the two.

- 10.2.5 Revise to include software, CM, and QA.

MIL-STD-481A, MIL-STD-482A, and DOD-STD-1467.

o No significant deficiencies noted.

MIL-STD-2168 (MIL-Q-2168) and the Joint Regulation.

o Not reviewed due to uncertain status; MIL-STD-2168 may be merged into DOD-STD-2167A.

MII-STD-1456 (MU)

o This standard should be replaced by DI-E-2035B.

PRODUCT #2: CSA HANDBOOK.

Discussion. The Configuration Management Subpanel investigated the implications and requirements for developing a common software CSA system. As part of this process, the subpanel developed some specific products, including a dictionary of proposed software CSA data elements, a guideline document for writing the technical specifications for a generic CSA system, and a contractual document that would allow a government contracting agency to obtain access to certain data in the contractor's CSA system. The panel discussed several alternatives concerning the ultimate disposition of these documents.

Conclusions. The subpanel reached unanimous agreement on the need for a wide range of technical aids for personnel engaged in the procurement or development of an automated software CSA system. They concluded that the documents they produced would form a good beginning of a "handbook" that could be developed and made available to government personnel. The handbook would address acquisition and procurement issues (RFP, SOW, etc.), essential data elements, report generation, architectural design issues (distributed/centralized, etc.), host transportability, use of commercially available tools, data exchange, and other related issues that should be considered in the process of developing an automated software CSA system.

COGENT FACTORS.

Solutions.

Near Term (Recommendation 4-3-08). Disseminate the documents developed by the subgroup to government activities for critical review and update. These documents could then be used as the

beginning of a comprehensive handbook for use in procuring or developing software CSA systems. It is recommended that:

1. The JLC PDSS subgroup initiate a critical review of the documents developed by the subpanel, which are contained in this report.

2. The JLC develop a military handbook, for use by DOD activities, covering all aspects of procuring, modifying, or developing an automated software CSA system.

Estimated Cost. $150K

Time to Implement. 18 months

Dependencies. None

Alternatives. Without the CSA handbook, CSA system development times and costs will increase, users will not have the benefit of lessons learned by others, interface problems will be more acute, and there will be a greater potential for the inconsistent application of DOD software configuration management practices.

Method of Implementation. The development effort should be assigned to the PDSS subgroup, who will manage the effort. The PDSS subgroup will issue development tasks as required.

Justification for Prioritization. The development of a CSA handbook would provide the following benefits:

1. For prospective users of common, commercially available CSA tools, the handbook would provide the information needed to adapt or otherwise tailor the available tools to their unique requirements.

2. For those involved in developing their own CSA system, the handbook will shorten both the procurement and development times, and significantly reduce overall costs.

3. Use of the handbook will promote the consistent implementation of DOD software configuration management guidelines, procedures, and practices across all the Services.

PRODUCTS.

All documents that were developed by the Configuration Management Subpanel are attached to this report. The following paragraphs provide rationale for the use of these documents.

DATA ELEMENT DICTIONARY. Adoption of a common set of data elements will make the following possible:

1. Cost effective transfer of CSA data from the developer to the software support activity during development and when the system is delivered.

2. The interchange of data and tools between software support activities.

Return on Investment. The Return on Investment (ROI) is the capability to cost-effectively transfer CSA data. The near term benefit is that software support activities will be able to use the CSA data that was created by the developer on their own CSA tools. The long term benefit is increased flexibility of options. It will be more cost effective for a software support activity to support multiple systems. It will also be possible to allow software support activities to share data or combine their data. Then, they can also share tools that can manipulate the standard data elements. An additional benefit is the installation of a framework for data sharing. Data structures may be increased in scope by the inclusion of software quality metrics data elements, human resource and planning data elements, and life cycle end-to-end costing data elements. The management and contractual framework would be in existence to perform cost-benefit analysis of various implementation and design approaches. The concept of reusable modules/units of code would be facilitated through the data base categorization of a unit through CSA data base parameters. A suitable unit of code may be located by defining the appropriate parameters and performing a data base query. Any deliverable, standard set of CSA data, constructed in accordance with the standard, may be accessed in the fashion most economical to that user. The cost to implement will vary with the level of sophistication posessed by the existing software support activities and principal development activities. However, by specifying only the data elements, structures, and delivered data format, the agencies implementing automated CSA are not precluded from utilizing existing capital assets expended for CSA. If the automated CSA conforms well to DOD-STD-2167, there will be minimal or no cost. If the CSA does not currently conform, it is expected that the activity will conform to that standard anyway.

Method of Implementation. The method of implementation is to include the data elements list in a CSA Handbook to receive widest dissemination possible.

Justification for Prioritization. This is the most important result of the subpanel. If common data elements and structures are kept, then diverse options are available to the government. It will become possible to develop common tools and share support software between the many diverse organizations involved

(procuring agency, developer, IV&V, PDSSA, etc.). The large market for tools, created due to the required standard data sets, will encourage private enterprise to develop off-the-shelf packages that solve the majority of the CSA automation problems (not unlike the industry interest in Ada support tools).

GUIDELINES FOR THE STATEMENT OF WORK. This product is a guideline that can be used for developing a SOW for a CSA system. It provides all of the sections that need to be included. The contents of each section consist of known requirements, recommendations for features that are "nice to have", and the trade-off considerations that need to be addressed. This guideline can be used by a developer or a PDSSA.

Near Term Solution. The full or partial implementation of this SOW can be done at any time, including near, mid and long term. If funds are not available for the whole system, a phased implementation plan can be developed to start with the mandatory requirements, and over time add the recommended features.

Return on Investment. The whole system could be developed cost-effectively in approximately twenty-four months. If there is an existing system, the development of the system would need to start with a one to three month study where the existing features would be compared to the SOW's recommendations. Once the differences are identified, the alternatives of augmenting the existing system can be costed. Then a total or a phased implementation plan can be developed. The cost of augmentation would vary with the degree of augmentation required, but would be less than the cost of a totally new development, which is estimated at one million dollars.

Dependencies. This SOW guideline can be used to design a SOW which can be implemented independently of any of the other products, but the adoption of the common set of data elements is recommended as a prerequisite. There could be a delay in the start of the implementation of the system if the common data set of data elements has not yet been identified.

List of Alternatives. The only alternative to automation of CSA is to track the data manually. While this alternative is not recommended by the panel, it may be a practical reality dictated by funding or schedule constraints. As long as the portions that are automated follow a total plan, then over time, the manual processes can be gradually automated.

Method of Implementation. Once again, it is recommended that this plan be included in a CSA Handbook.

DRAFT REQUIREMENT FOR CSA ACCESS.

This product is draft language for inclusion in a SOW in a RFP package for development of an MCCR system. It is designed to provide the program manager with visibility into the developer's CSA system and to provide the PDSSA with an effective transfer of the developer's CSA data at the time that the system is delivered to the government.

Near Term Solution. This language can be immediately included in RFPs. The program manager for an existing development may want to consider adding this as a contract modification if the program is still in the early stages.

Return on Investment. The ROI is twofold. The visibility into the development process will be greatly enhanced. This benefit will be realized as soon as the systems that are contracted with this language reach the stage of development where data items are being created (a time frame of under a year from the drafting of the RFP until the first delivery of a data item). The second benefit will be that the PDSSA can use the developer's CSA data. This benefit will not be realized until systems that include this contract language are delivered (a minimum of two years).

The cost of the access to the developer's CSA program will vary greatly depending on the differences between the developer's and the government's designated target computers. It is estimated that the range will be from $10K (for fully compatible systems) to a worst case of $100K (for very incompatible systems). This estimate is per interface; there may be a requirement for more than one interface (e.g., to IV&V and the PDSSA). Implementation of this automated interface will not extend the total program development time, as it can be done in parallel with the early development activities. The duration of the development will range between approximately one and six months.

Dependencies. This product will be more effective if standard data elements are utilized, but that is not a prerequisite. There are no mandatory dependencies for this product.

Method of Implementation. The method of implementation is the inclusion of the draft RFP language in the handbook for CSA system development.

Justification for Prioritization. The primary justification is the great difficulty experienced by the PDSA's when a system is delivered to the government. They must undergo an expensive and protracted process to incorporate the delivered configuration items into their CSA System. The cost savings incurred if this hand-off is automated, should offset the costs incurred by creation of an automated interface.

PRODUCT #3 COMMON AUTOMATED CSA SYSTEM

Discussion. The Configuration Management Subpanel investigated the implications and requirements for developing a common software CSA system. Issues addressed included the exchange of data among CSA systems, the transfer of software CSA data from a developing activity to a PDSS activity, CSA report formats, and the trade-offs of various CSA system architectural approaches.

Conclusions. The subpanel reaffirmed the recommendation of the Orlando I Configuration Management Subpanel, that a common software CSA system be developed. This system would automate the software configuration management functions required by DOOD 5010.19, DOD-STD-2167, DOD-DTD-2167A, and related standards. This system would be available for use by government activities, and would be available as Government Furnished Equipment (GFE) to contractors working on government software projects. The system should be developed from existing Service baselines to the extent practicable, and should consist of building blocks that may be replaced with commercial software tools already in place at PDSS activities. The system must be extensible and user tailorable to local site or project unique requirements, such as report formats, terminology, and security classification, and provide for the exchange of data among CSA sites. The system must support multiple site, multiple project, multiple host, and multiple participant configuration management activities from programmers to project managers. After development, the system could be turned over to one of the Service PDSS Software Commonality Offices proposed by the Orlando II Software Technology Transition Panel.

COGENT FACTORS.

Solutions.

Near Term. No adequate near term solution. If approved, the CSA Handbook would provide initial guidance for those engaged in developing their own CSA system.

Mid Term (Recommendation 4-3-09). It is recommended that a DOD owned, common automated software CSA system be made available.

o The JLC support the development of a common automated CSA system. This recommendation involves two complementary actions:

1. The JLC fund the development of a formal system specification for the CSA system.

2. The JLC sponsor, promote, and oversee the development of the CSA system. In this capacity, the JLC will solicit development funds from prospective user activities.

Estimated Cost.	$100K	(JLC)
	$1M	(Other)
Time to Implement.	1 year	(System Specification)
	2 years	(CSA system)

Dependencies. Development funding must be obtained from potential users. However, the chances of obtaining the development funds will be greater if the JLC approve the seed money to begin work on the formal system specification.

Alternatives. Without a common set of software CSA tools, the overall DOD funding needed for Service activities to independently develop and maintain their own systems will increase astronomically, unnecessary time will be consumed, training costs will soar, it will be increasingly difficult to transfer data among the various CSA systems, and there will be a greater potential for inconsistent implementations of DOD software configuration management practices.

Method of Implementation. The development effort should be assigned to the PDSS Subgroup, who will manage the effort. The PDSS Subgroup will issue development tasks as required.

Justification for Prioritization. The development of a common automated software CSA system would provide the following benefits:

1. Significantly reduced overall government software development and maintenance costs. The availability of an adaptable set of integrated software CSA tools would save considerable R&D development funds. Also, the reduction in the number of multiple, functionally equivalent, CSA systems would significantly reduce software life cycle support costs.

2. Provide for a valid and consistent implementation of DOD software configuration management requirements across all the Services.

3. The use of common CSA tools and procedures will significantly reduce overall training requirements.

4. The use of common CSA tools, with data import/export features, will foster the exchange of data among user sites and provide a needed capability for remote site backup of critical information.

PRODUCTS.

There are no specific products associated with this recommendation.

(Intentionally Blank)

PANEL IIIB - CONFIGURATION MANAGEMENT

Chairman
Owen McOmber
COMPTEK Research, Inc.
2929 Canon Street, Suite 200
San Diego, CA 92106
(619) 225-9921

Subgroup IIIB(1): Software Configuration Management Policy

Configuration Management is a critical software support function that has the potential for significant cost avoidance if effective implementing standards and common tools were to exist among industry and each Service. DOD CM methods must correctly reflect the unique nature of software configuration items.

OBJECTIVE.

Identify software and firmware related deficiencies in DOD configuration management directives and standards, and develop a recommended approach for implementing required changes.

Task. Review current DOD CM directives and standards and identify software and firmware related deficiencies. Review will be based on:

- o General correctness and currency
- o Software related CM requirements
- o Consistency with DOD-STD-2167/2167A
- o PDSS requirements

Review Documents:

DODD 5010.19 - Configuration Management

Joint CM Regulation (1 July 1974)

Department of the Army.....AR 70-37
Department of the Navy.....NAVMATINST 4130.1A
Marine Corps.....MCO 4130.1A
Department of the Air Force....AFR 65-3
Defense Supply Agency.....DSAR 8250.4
National Security Agency.....NSA/CSS 80-14
Defense Communications Agency..DCAC 100-50-2
Defense Nuclear Agency.....DNA INST 5010.18

Joint CM Regulation (30 October 1981 Draft)

MIL-STD-483A - Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs.

DOD-STD-480A - Configuration Control - Engineering Changes, Deviations, and Waivers

MIL-STD-481A - Configuration Control - Engineering Changes, Deviations, and Waivers (Short Form)

MIL-STD-1456 - Contractor Configuration Management Plans

MIL-STD-482A - Configuration Status Accounting Data Elements and Related Features

MIL-STD-490A - Specification Practices

MIL-STD-1521B - Technical Reviews and Audits for Systems, Equipment, and Computer Programs

DOD-STD-2167 - Defense System Software Development

DOD-STD-2167A - Defense System Software Development (Draft)

DOD-STD-1467 - Software Support Environment

MIL-STD-499 - Engineering Management

PRODUCT.

Report on software and firmware related deficiencies in current DOD CM directives and standards, including proposed new methods, practices, and procedures, with implementation recommendations.

Subgroup Members:

John Benson	Airborne Software
Ronald Berlack	Sanders Associates
Carl Berry	Defense Data Mgmt. Off.
Perry DeWeese	Lockheed Georgia
Kris Hatakeyama	NSWSES
John Holovet	Army Material Command
Frank Hubans	General Dynamics
Clyde Kluge	OC-ALC/MMECT
Maj Ken Miller	AFSC
Dennis Nickle	E-Systems
1stLt Gerald Schumacher	HQ AFLC/MMEEE

Subgroup IIIB(2): Software Configuration Status Accounting

OBJECTIVE.

Develop basic procurement documents for the development of an automated standard software CSA system.

Tasks. Review and analyze Government Furnished Information (GFI) in the areas of:

1. Defining essential software CSA data elements.
2. Centralized/distributed processing issues.
3. Data exchange protocols/approaches.
4. Report generation methods/formats.

Identify and analyze other issues relative to automating software configuration status accounting.

PRODUCT.

High level requirements document for an automated standard software configuration status accounting system. Include alternative options if appropriate.

Subgroup Members:

Robert Both	CECOM
Robert Havey	DOD Tech. Analysis Office
Claire Lohr	Software Systems Corp
Capt Tony Romero	MCTSSA

PANEL IIIB BIBLIOGRAPHY & ABSTRACT.

DODD 5010.19 - Configuration Management

Joint CM Regulation (1 July 1974)

Department of the Army.....AR 70-37
Department of the Navy.....NAVMATINST 4130.1A
Marine Corps.....MCO 4130.1A
Department of the Air Force....AFR 65-3
Defense Supply Agency.....DSAR 8250.4
National Security Agency.....NSA/CSS 80-14
Defense Communications Agency..DCAC 100-50-2
Defense Nuclear Agency.....DNA INST 5010.18

Joint CM Regulation (30 October 1981 Draft)

MIL-STD-483A Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs.

DOD-STD-480A Configuration Control - Engineering Changes, Deviations, and Waivers

MIL-STD-481A Configuration Control - Engineering Changes, Deviations, and Waivers (Short Form)

MIL-STD-1456 Contractor Configuration Management Plans

MIL-STD-482A Configuration Status Accounting Data Elements and Related Features

MIL-STD-490A Specification Practices

MIL-STD-1521B Technical Reviews and Audits for Systems, Equipment, and Computer Programs

DOD-STD-2167 Defense System Software Development

DOD-STD-2167A Defense System Software Development (Draft)

DOD-STD-1467 Software Support Environment

MIL-STD-499 Engineering Management

Defense Department Configuration Management Report, Comptek Research, Inc., 24 January 1987

Configuration Management for Mission Critical Software: The Los Alamos Solution, G. Cort and D. M. Barrus, Los Alamos National Laboratory (undated)

CECOM Life Cycle Software Engineering Center Status Accounting Data Elements, U. S. Army CECOM, Fort Monmouth, NJ

Unit Development File Requirements, 29 January 1985

Configuration Management Reporting, CMS Standard Reports Manual

ISEA Subsystem Configuration Status Accounting Report (CSAR) User's Guide

Technical Specifications for a CECOM Life cycle Software Engineering Software Change Control Automation System

---

Title. DODD 5010.19

Subject. Configuration Management

Date. 1 May 1979

Status. Being updated.

Abstract. This is the top level CM directive within DOD. It defines policies for the configuration management of materiel including systems, equipment, computer programs, facilities, and other designated items throughout their life cycle. It also includes the following top level definitions:

1. Configuration Management. The engineering management procedures that include the following:

2. Configuration Identification. Selection of the documents which identify and define the configuration baseline characteristics of an item.

3. Configuration Control. Controlling changes to the configuration and its identification documents.

4. Configuration Status Accounting. Recording and reporting the implementation of changes to the configuration and its identification documents.

5. Configuration Audit. Checking an item for compliance with the configuration identification.

The directive requires that the degree of CM applied for an item shall be appropriately tailored to be consistent with the complexity, size, quantity, intended use, mission criticality, and the life cycle phase of the item. (see DODD 4120.21 below)

The directive addresses the CM of interface baseline characteristics, items developed wholly or partially with government funding, items developed with private (industry) funding, and application of CM during the various phases of development and/or acquisition.

The directive tasks DCMC with conducting the DOD Configuration Management Program, and designates the Navy as lead Service for configuration management. In this role, the Navy is responsible for maintaining the Joint Configuration Management Regulation and other related joint documents, and for chairing the DCMC.

Title. DODD 4120.21

Subject. Application of Specifications, Standards, and Related Documents in the Acquisition Process.

Date. 3 November 1980

Status. Current

Abstract. This directive governs the application and tailoring of specifications, standards, and related documents that can be cited in defense contracts (e.g., MIL-STD, MIL-SPEC) but not directives, instructions, and regulations, etc., which can no longer be cited by reference in contracts. This directive is not directly related to the configuration management policy chain, but it is indirectly related because it is cited in DODD 5010.19 as applicable to military standards and specifications covering configuration management.

This directive is significant because, among other things, it includes the top level definition of "tailoring", which is defined as:

> "The process by which individual requirements (sections, paragraphs, sentences) of the selected specifications, standards, and related documents are evaluated to determine the extent to which they are most suitable for a specific system and equipment acquisition, and the modification of these requirements to ensure that each achieves an optimal balance between operational needs and cost."

Tailoring of data requirements shall consist only of the exclusion of those sections, paragraphs, or sentences in an approved document's information requirement or Data Item Description.

This directive also authorizes the publication of DOD 4120.21M, a manual on the application of specifications, standards, and related documents in the acquisition process.

Title. Joint DOD Services/Agency Regulation

Subject. Configuration Management

Date. 1 July 1974

Status. Scheduled to be updated to DODI 5010.XX

Abstract. This is a relatively detailed joint regulation that prescribes uniform policies and guidance for the DOD components responsible for implementing CM within DOD. This regulation is applicable to the following DOD components, as implemented by the indicated documents:

Army.....AR-37
Navy.....NAVMATINST 4130.1A
Air Force...AFR 65-3
USMC.....MCO 4130.1A
DSA.....DSAR 8250.4
NSA.....NSA/CSS 80-14
DCA.....DCAC 100-50-2
DNA.....DNA INST 5010.18

A more detailed draft revision of this regulation was published and partially coordinated in 1981. The Navy was the only DOD component to implement the new revision (in 1983), but only within the Naval Materiel Command (NMC). Due to the resistance of OSD and the other DOD components, the new revision was not widely accepted or used in the Navy, particularly after the NMC was dissolved in 1985.

The provisions of the regulation apply to major defense systems, systems requiring Service/agency decision processing, and selected end items for reason of system integration or interface control. The regulation states that application of the cited management requirements is not intended to impose any special organization, structure, or organizational title, particularly upon contractors.

The regulation states that the CM process shall be tailored to the quantity, size, scope, stage of life cycle, nature, and complexity of the configuration item involved. The selection of configuration items to be managed is determined by the government's need to control an item's inherent characteristics or to control the item's interface with other items. Exploratory and advanced development efforts including prototype projects are exempt from application of formal configuration management unless such application is advantageous to management of the project.

The regulation describes the extent to which CM is applied during the various life cycle phases of the configuration item. It requires that configuration identification be established in the form of technical documentation, and that software configuration identification be applied to the actual computer programs as well as their associated documentation. Procedures are defined for applying configuration management to joint Service/agency development projects. The regulation also requires that provisions for CM shall be included in all contracts or in-house equivalents (interagency agreements, task orders, and other such tasking to government development activities). It requires that configuration baselines be employed throughout the life cycle of a configuration item to ensure an orderly transition from one major commitment point to the next in the system engineering, production, and logistic support processes.

Title. MIL-STD-483A

Subject. Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs.

Date. 10 May 1985

Status. Interim version

Abstract. This standard sets forth CM practices which are to be tailored to specific programs implemented by reference in the SOWs. This standard also establishes CM requirements that are not covered in DOD-STD-480A, MIL-STD-481, MIL-STD-482, and MIL-STD-490.

This is an interim standard that was informally coordinated with DOD-STD-2167. It has been conditionally approved, but must be coordinated with the DCMC and agreed upon by all concerned.

Title. MIL-STD-480A

Subject. Configuration Control - Engineering Changes, Deviations and Waivers

Date. 29 December 1978

Status. Needs updating

Abstract. This standard establishes the requirement for making an overall assessment of all proposed changes to established configuration items. It contains detailed instructions for submitting ECPs and related information, and requires that all known interface effects throughout the baseline be considered in the change analysis process. The standard imposes on the

contractor the responsibility for analysis of overall system impact, including fiscal and integrated logistic support effects.

Title. MIL-STD-481A

Subject. Configuration Control - Engineering Changes, Deviations and Waivers (Short Form)

Date. 18 October 1972

Status. Needs updating.

Abstract. This standard establishes the requirement and provides instruction in the preparation and submittal of abbreviated ECPs. Information submitted emphasizes the impact on the item under contract, with a limited description of the effect on interfaces and integrated logistic support. This standard is intended for use in contracts involving the procurement of multiple items or items for which the prescribed detailed design was not developed by the present contractor.

Title. MIL-STD-1456

Subject. Contractor Configuration Management Plans

Date. 10 May 1972

Status. Needs updating

Abstract. This is an Army military standard that establishes the format and contents of CM plans prepared by contractors. It does not address software configuration management.

Title. MIL-STD-482A

Subject. CSA Data Elements and related Features

Date. 1 April 1974

Status. Needs updating

Abstract. This standard prescribes standard status accounting data elements, interim data elements, and their related data items, codes, use identifiers, and data chains (related features). The data elements established by the standard are to satisfy the requirement for all CSA records prepared by or for DOD components in accordance with DODD 5010.19. It does not address software related issues adequately.

Title. MIL-STD-490A

Subject. Specification Practices

Date. 4 June 1985

Status. Interim version, conditionally approved

Abstract. This military standard sets forth practices for the preparation, interpretation, change, and revision of program peculiar specifications prepared by or for the DOD components. The standard provides a systematic hierarchy of specification models ranging from high level specifications to the detailed specifications. The purpose of the standard is to establish uniform practices for specification preparation, to ensure the inclusion of essential requirements, and to aid in the use and analysis of specification content.

This is an interim standard that was informally coordinated with DOD-STD-2167. It has been conditionally approved, but must be coordinated with the DCMC and agreed upon by all concerned.

Title. MIL-STD-1521B

Subject. Technical Reviews and Audits for Systems, Equipment, and Computer Software

Date. 4 June 1985

Status. Interim version, conditionally approved

Abstract. This standard prescribes the requirements for the conduct of technical reviews and audits on systems, equipment, and computer software. Reviews and audits covered include:

System Requirements Review (SRR)
System Design Review (SDR)
Software Specification Review (SSR)
Preliminary Design Review (PDR)
Critical Design Review (CDR)
Test Readiness Review (TRR)
Functional Configuration Audit (FCA)
Physical Configuration Audit (PCA)
Formal Qualification Review (FQR)
Production Readiness Review (PRR)

This is an interim standard that was informally coordinated with DOD-STD-2167. It has been conditionally approved, but must be coordinated with the DCMC and agreed upon by all concerned.

TECHNICAL SPECIFICATIONS GUIDELINES FOR A SOFTWARE CHANGE CONTROL AUTOMATION SYSTEM

INTRODUCTION.

This document provides guidelines for the technical specifications for a Software Change Control Automation System (SCCAS). SCCAS is a computer program, or an integrated set of computer programs, that automates the management and control of changes for computer based files of information. SCCAS will provide a repository (the Configuration Data Base) for managed files, an access control scheme to limit the availability of files, and automated procedures for tracking authorized changes to files and reporting documents that may require change based on dependencies on other files.

ASSOCIATED EQUIPMENT.

[Consideration: The SCCAS host computer environment shall be specified in the contract. If applicable, the following requirements shall also be specified in the contract:

- o SCCAS transportability (system, data, or both).
- o Support of multiple programs.
- o Support of multiple sites.]

APPLICABLE DOCUMENTS.

DOD-STD-2167
DOD-STD-483
DOD-STD-480
DOD-STD-482
DOD-STD-1521

GENERAL REQUIREMENTS.

Software Interdependencies. The SCCAS software, as installed, must constitute a complete software system. Other than the host operating system, SCCAS shall not require the presence of any additional software to perform the functions specified in this document.

[Consideration: Automation or nonautomation of specific functions of SCCAS will depend on the flow of data and management control at the using activity, and on cost considerations. The use of existing systems, off-the-shelf software, a custom system, or some combination of these will depend on the equipment and software currently in use, the availability of suitable

off-the-shelf software, and cost considerations. The use of existing software to supplement the SCCAS should be considered.]

Training Support.

[Consideration: At the time of contract the required form and amount of training support shall be specified (e.g., video tape, interactive on-line-tutorial at contractor site, contractor provided training at the contractor or delivery site).]

Software Updates, Warranties and Support.

[Consideration should be given to the type and term of warranty, the deliverability of updates of the system over a specified period of time and vendor support to the system for some specified period of time.]

Data Base Requirements.

This component of the SCCAS acts as a repository for all data managed by the configuration management tool. The data shall include software source, object, documentation and CM files.

[Consideration: At the two extremes of status accounting utilization there are the following methodologies:

Distributed. Every development and maintenance programmer works directly with the status accounting data base, using the SCCAS commands and host facilities to accomplish all programming and maintenance activities.

Centralized. At this extreme, the programming and maintenance staff do not interact with the status accounting data base but conduct their programming activities externally. A manager is then responsible for collecting status accounting information from each of the users and centrally storing this information.

Some of the advantages and disadvantages of each approach are listed as follows:

Distributed Advantages. Allows high level of visibility throughout the development process. Provides capability to identify software version changes with a high degree of resolution.

Distributed Disadvantages. Overhead requirement placed on development personnel depends on the transparency of the status accounting system implementation. An additional requirement for a database administrator to define and maintain access control for every SCCAS user is created. Developers must now become proficient with an additional software/hardware interface.

Centralized Advantages. Access control problems are eliminated. Programmer overhead is reduced.

Centralized Disadvantages. Increased maintenance effort required to export modules from the status accounting data base to the program maintenance data base. Timeliness of baseline status information is degraded.

Data Base Organization.

[Recommendation: The data base should support structured software relationships as defined in DOD-STD-2167, paragraph 4.2.]

Data Structures. SCCAS shall allow data elements to be accessed by any data structure of the data base.

The data base must be capable of starting and identifying complete versions of the data structure, and identifying all changes between any two versions of the data structure.

Data Base Access Control.

[Considerations: A decision must be made regarding the applicable level and type of data base access control (e.g., Physical, Operating System (OS) supported password security or OS independent access limitation).

SCCAS can maintain a list of the users authorized to access specific data elements and/or data structures of the data base.

Copies can be used to protect the integrity of the controlled SCCAS data base.]

[Recommendations:

Access Levels. SCCAS should incorporate a mechanism to define specified levels of access for authorized users of a specified data structure.

User Access. SCCAS should permit an individual with user access rights to a data structure to read, or read and modify, the components of that data structure or the components of any subordinate data structure.

SCCAS must recognize at least two different categories of user access rights: "read only" and "read and modify".]

Archiving. SCCAS must be capable of archiving data structure versions. The archiving mechanism must retain copies of all old versions of a modified data structure.

[Consideration: Control over how many previous versions are actually archived, where they are stored, frequency of archive, and recall, must be decided. Archive control must be implemented to allow for local management of the system.]

Data Structure and Data Elements.

The data dictionary must be specified in detail. SCCAS shall allow deletions, changes, or additions to the data dictionary.

USER INTERFACES.

User Interface. SCCAS shall provide a user interface that allows manipulation of files and data structures in the data base.

[Considerations: Interaction Modes. In order to accommodate users whose experience with the tool may vary from novice to expert, SCCAS may incorporate the capability to be adjusted to the user's level of expertise and be adapted to interact with the user at that level. To achieve this goal, the interface can allow the user to specify commands in various different formats and can, if necessary, provide the user with additional prompts and/or other information about the command during the command specification operation.

A decision must be made regarding whether SCCAS will be command driven, menu driven, or both, and whether to allow the creation and use of user specified menus and macros.]

[Recommendations:

Provide for interactive interface.

Command Definitions: In order to simplify operation a command driven system should use the same set of commands at each level of the data base hierarchy.

HELP Facility: SCCAS should incorporate an interactive HELP facility to provide the user with information relating to command syntax and usage. SCCAS must allow the HELP facility to be invoked at any point during the dialogue with a user. In particular, SCCAS should allow the HELP facility to be invoked from within a command line.]

[Recommended User Command Functions: SCCAS shall provide standard commands to perform the following user functions:

a. Modify a file.

b. Combine files or data structures.

c. Copy files or data structures.

d. Logically delete previously made changes.

e. Logically delete a file or data structure.

f. Place a copy of a file outside the data base.

g. Admit a new version of a file.

h. Create a new file or data structure.

i. List a file or data structure.

j. List changes to a file or data structure.

k. List differences between files or data structures.

l. Rename a file or data structure.

m. Change password.

n. Define access specifications.

o. Modify an existing access specification.

p. List access specifications.

q. Physically delete files or data structures.

r. Set database storage characteristics.

s. Modify database storage characteristics.

t. List database storage characteristics.

u. Generate reports.

v. Store reports.

w. Print reports.

Reports. SCCAS shall provide the capability to extract standard and locally defined reports on the data base.

[Consideration: Reports such as those described below should be built into the system.

Structure Report. A structure report should detail the level structure of configured systems, including all relevant file and change information.

Status Accounting. A configuration status accounting report shall describe the structure as of specific dates/releases and identify the consistent engineering change revision levels for each data structure in the data base. Textual descriptions of the changes for each data structure in the data base shall be maintained and selectively incorporated into this report.

Differences. A differences report shall provide for a specified system or structure the specific differences between any two versions of a file or set of files at a level. This will include differences in validation test data files.

User Definable Reports. The following reports shall be definable based on the data base information:

a. Problem summary and status.

b. Change summary and status.

c. Data Extraction from input forms (field selection by multiple search criteria).

d. User specific report subsets.]

[Recommendation: Use sample formats as shown in Appendix VIII of MIL-STD-483A.]

System Failure Recovery Facility.

[Recommendations:

Journaling. SCCAS should incorporate a journaling feature that records all operations made on the data base since the last backup, if those operations have changed the data base.

Fault Recovery. SCCAS should provide the capability to automatically rebuild the data base from the journal file. Subsequent to a hardware or software fault that compromises the integrity of the database, SCCAS should allow managers the option to restore the data base to its default configuration.]

Security. The level of classification of the software and data to be managed by SCCAS must be considered when implementing SCCAS.

[Recommendation: To simplify operations with respect to security, all data on the system should be managed at a "system high" security level.]

Other Considerations. Integration of SCCAS with a comprehensive CM system and with the software support environment should be considered when implementing SCCAS.

CONTRACT STATEMENT FOR DATA ELEMENTS REQUIRED FOR CSA TO BE INCLUDED IN THE RFP FOR A SYSTEM CONTAINING SOFTWARE

The contractor shall maintain an automated data base throughout development consisting at a minimum of the data elements shown in Table 1. [Table 1. Required CSA Data Elements should be used until MIL-STD-482 is updated to accommodate the required data elements. In either case the list must be tailored to accommodate the specific procurement. The procuring activity must specify the level of visibility into the software and the level of detail of the required data base in the RFP.]

On line access to the data base shall be provided to the government PM or his designated representative throughout the development. Provisions shall be made to allow the designated government representative to read, and down load, the latest update of the data base at any time on the equipment specified in the SOW. [The equipment being used by the PDSSA or the IV&V activity and the required interfaces and formats should be specified either here or in another attachment.] [Consideration should be given to the required frequency of government access and whether continuous government access is necessary.]

The contractor shall present his approach to transferring the data from his facility to the facility designated by the government in his response to the RFP. The contractor shall demonstrate this data conversion prior to delivery of the first software related document required in the CDRL. ["The first software related document" should be replaced by a specific document in the RFP.] [A CDRL - DD Form 1423 - is needed to support the data delivery.]

A machine loadable copy of the data base shall be delivered to the government prior to the FCA/PCA. Data base copies shall be delivered in accordance with the CDRL and shall be loadable on the equipment specified in [the equipment being used by the PDSSA and the delivery medium and format should be specified either here or in another attachment.]

(Intentionally Blank)

**********DATA SETS**********

Computer Software Configuration Item Data Set

Development Record Data Set

Software Requirements Specification Data Set

Interface Requirements Specification Data Set

Software Product Specification Data Set

Software Top Level Design Document Data Set

Software Detailed Design Document Data Set

Software Test Plan Data Set

Interface Design Document Data Set

Data Base Design Document Data Set

Test Documentation Data Set

Software Test Description Data Set

Software Test Procedure Data Set

Manuals Data Set

Computer System Operators Manual Data Set

Computer System Users Manual Data Set

Reviews and Audits Data Set

Version Description Document Data Set

Configuration Control Management Data Set

Notice of Revision Data Set

Specification Change Notice Data Set

Engineering Change Proposal Data Set

Software Problem/Change Report Data Set

Software Trouble Report Data Set

Request for Deviation/Waiver Data Set

System Information Data Set

Computer Software Configuration Item Data Set

System Version Data Set

Software Change Order Data Set

Alteration Data Set

Patch Data Set

**********DATA ELEMENTS**********

COMPUTER SOFTWARE CONFIGURATION ITEM DATA SET - DATA ELEMENTS

ISSUING AGENCY DATA ELEMENT
DOCUMENT NUMBER DATA ELEMENT
CONTRACT NUMBER DATA ELEMENT
CONTRACT DOCUMENT REQUIREMENTS LIST ITEM NUMBER DATA ELEMENT
INDEX DATE DATA ELEMENT
COMPUTER SOFTWARE CONFIGURATION ITEM NOMENCLATURE DATA ELEMENT
SYSTEM TITLE DATA ELEMENT
SYSTEM NUMBER DATA ELEMENT
INDEX ISSUE NUMBER DATA ELEMENT
TABLE OF CONTENTS DATA

COMPUTER SOFTWARE CONFIGURATION ITEM DATA SET

DEVELOPMENT RECORD DATA SET -- DATA ELEMENTS

COMPUTER SOFTWARE CONFIGURATION ITEM NUMBER
COMPUTER SOFTWARE CONFIGURATION ITEM NOMENCLATURE
DATE ISSUED - SOFTWARE REQUIREMENTS SPECIFICATION
DATE ISSUED - INTERFACE REQUIREMENTS SPECIFICATION
DATE OF AUTHENTICATION - INTERFACE REQUIREMENTS SPECIFICATION
DATE OF AUTHENTICATION - SOFTWARE TEST REPORT
DATE OF AUTHENTICATION/APPROVAL - SOFTWARE REQUIREMENTS SPECIFICATION
DATE OF ISSUE - DATA BASE DESIGN DOCUMENT
DATE OF ISSUE - INTERFACE DESIGN DOCUMENT
DATE OF ISSUE - SOFTWARE TOP LEVEL DESIGN DOCUMENT
DATE OF ISSUE - SOFTWARE TEST PLAN
DATE OF ISSUE - SOFTWARE DETAILED DESIGN DOCUMENT
DATE OF ISSUE - SOFTWARE TEST DESCRIPTION
DATE OF ISSUE - SOFTWARE TEST PROCEDURE
DATE OF ISSUE - SOFTWARE TEST REPORT
DATE OF ISSUE - SOFTWARE PRODUCT SPECIFICATION
DATE OF APPROVAL - SOFTWARE TEST PLAN
DATE OF APPROVAL - SOFTWARE TEST PROCEDURE
START DATES - SOFTWARE SPECIFICATION REVIEWS
START DATES - PRELIMINARY DESIGN REVIEW
START DATES - CRITICAL DESIGN REVIEW
START DATES - TEST READINESS REVIEW
START DATES - FUNCTIONAL CONFIGURATION AUDIT
START DATES - PHYSICAL CONFIGURATION AUDIT
STOP DATES - SOFTWARE SPECIFICATION REVIEWS
STOP DATES - PRELIMINARY DESIGN REVIEW
STOP DATES - CRITICAL DESIGN REVIEW
STOP DATES - TEST READINESS REVIEW
STOP DATES - FUNCTIONAL CONFIGURATION AUDIT
STOP DATES - PHYSICAL CONFIGURATION AUDIT
CONTRACTOR NAME
CONTRACT NUMBER

COMPUTER SOFTWARE CONFIGURATION ITEM DATA SET

ALL SPECIFICATION DATA SETS -- GENERIC DATA ELEMENTS

SPECIFICATION TYPE
SPECIFICATION TITLE
SPECIFICATION VOLUME NUMBERS
SPECIFICATION APPENDICES NUMBERS
ISSUE
SOFTWARE CHANGE NOTICE NUMBER
SOFTWARE CHANGE NOTICE CHANGE TYPE
CHANGE TITLE
CHANGE DATE
APPROVED CHANGES
ENGINEERING CHANGE PROPOSAL NUMBER
ENGINEERING CHANGE PROPOSAL TITLE
APPLICABLE VOLUME
APPLICABLE APPENDIX
APPROVAL DATE

VERSION DESCRIPTION DOCUMENT DATA SET -- DATA ELEMENTS

SYSTEM IDENTIFICATION NUMBER
VERSION DESCRIPTION DOCUMENT NUMBER
EXTERNAL VERSION DESCRIPTION DOCUMENT NUMBER
TITLE
DESCRIPTION
DRAFT DATE
FINAL DATE
PROJECT LEADER APPROVAL
CONFIGURATION MANAGEMENT APPROVAL
QUALITY EVALUATION APPROVAL
PROJECT MANAGEMENT APPROVAL

CONFIGURATION CONTROL MANAGEMENT DATA SET

NOTICE OF REVISION DATA SET -- DATA ELEMENTS

SYSTEM IDENTIFICATION NUMBER
NOTICE OF REVISION NUMBER
EXTERNAL NOTICE OF REVISION NUMBER
INCOMING OR OUTGOING
TITLE
ORIGINATOR
ORIGINATOR ADDRESS
ORIGINATOR PHONE
DESCRIPTION
RELATED ENGINEERING CHANGE PROPOSAL NUMBER
RELATED COMPUTER SOFTWARE CONFIGURATION ITEM
RELATED COMPUTER SOFTWARE CONFIGURATION ITEM VERSION
DATE PREPARED
DATE LOGGED
IN-HOUSE RELEASE DATE
DATE TO PROJECT LEADER

CONFIGURATION CONTROL MANAGEMENT DATA SET

SPECIFICATION CHANGE NOTICE DATA SET -- DATA ELEMENTS

SYSTEM IDENTIFICATION NUMBER
SOFTWARE CHANGE NOTICE NUMBER
ORIGINATOR
ORIGINATOR OFFICE SYMBOL
ORIGINATOR ACTIVITY CODE
DOCUMENT AFFECTED
COMPUTER SOFTWARE CONFIGURATION ITEM INVOLVED
RELATED SOFTWARE CHANGE NOTICE NUMBER
ORIGINATING ENGINEERING CHANGE PROPOSAL NUMBER
PRELIMINARY OR FINAL
DATE PREPARED
DATE APPROVED
UPDATE COMPLETE

CONFIGURATION CONTROL MANAGEMENT DATA SET

ENGINEERING CHANGE PROPOSAL DATA SET -- DATA ELEMENTS

DATE ENTERED
DATE LOGGED
SYSTEM IDENTIFICATION NUMBER
INTERNAL ENGINEERING CHANGE PROPOSAL NUMBER
ENGINEERING CHANGE PROPOSAL NUMBER
ENGINEERING CHANGE PROPOSAL DESIGNATION
MODEL
TYPE
ENGINEERING CHANGE PROPOSAL TYPE
ENGINEERING CHANGE PROPOSAL REVISION NUMBER
ENGINEERING CHANGE PROPOSAL CORRECTION NUMBER
ENGINEERING CHANGE PROPOSAL CLASSIFICATION
DATE ORIGINATED
ORIGINATOR NAME
ORIGINATOR ADDRESS
ORIGINATOR PHONE
ENGINEERING CHANGE PROPOSAL JUSTIFICATION CODE
ENGINEERING CHANGE PROPOSAL PRIORITY
CHANGE TITLE
NEED FOR CHANGE
DESCRIPTION OF CHANGE
PRODUCTION EFFECTIVITY
CHANGE IN SCHEDULE
NUMBER OF MONTHS
EFFECT ON SCHEDULE
COST IN $
OTHER COMPUTER SOFTWARE CONFIGURATION ITEM'S AFFECTED
  (FOR EITHER PRELIMINARY OR FINAL ENGINEERING CHANGE PROPOSAL)
RELATED NOTICE OF REVISION (INCOMING)
RELATED ENGINEERING CHANGE PROPOSAL
RELATED SOFTWARE PARENT CHANGE REQUEST
PARENT CHANGE FORM TYPE

PARENT CHANGE FORM NUMBER
ENERVATED ALTERATION NUMBER
RELATED REQUEST FOR DEVIATION
  (FOR ENGINEERING CHANGE PROPOSAL ONLY)
RELATED SOFTWARE CHANGE ORDER
  (STATUS ATTRIBUTES)
DISTRIBUTION DATE TO DIVISION CHIEF
DISTRIBUTION DATE TO PROJECT LEADER
DISTRIBUTION DATE TO CONFIGURATION MANAGEMENT
DISTRIBUTION DATE TO QUALITY EVALUATION
DISTRIBUTION DATE TO CONTRACTING OFFICER
DISTRIBUTION DATE TO DIRECTOR
APPROVAL/COMMENTS RECEIVED FROM DIVISION CHIEF
APPROVAL/COMMENTS RECEIVED FROM PROJECT LEADER
APPROVAL/COMMENTS RECEIVED FROM CONFIGURATION MANAGEMENT
APPROVAL/COMMENTS RECEIVED FROM QUALITY EVALUATION
APPROVAL/COMMENTS RECEIVED FROM CONTRACT OFFICER
APPROVAL/COMMENTS RECEIVED FROM DIRECTOR
  (SOFTWARE CONFIGURATION REVIEW BOARD ATTRIBUTES;)
ENGINEERING CHANGE PROPOSAL APPROVAL AUTHORITY
ENGINEERING CHANGE PROPOSAL APPROVAL DATE
ENGINEERING CHANGE PROPOSAL DISAPPROVAL DATE
ENGINEERING CHANGE PROPOSAL DEFERRED DATE
ENGINEERING CHANGE PROPOSAL ACTION DATE
ENGINEERING CHANGE PROPOSAL AUTHORIZING AGENT
ENGINEERING CHANGE PROPOSAL AUTHORIZING AGENT TITLE
ENGINEERING CHANGE PROPOSAL CONTRACT NUMBER
ENGINEERING CHANGE PROPOSAL CONTRACT LINE NUMBER
SOFTWARE CONFIGURATION REVIEW BOARD MEETING DATE
SOFTWARE CONFIGURATION REVIEW BOARD RECOMMENDATION
SOFTWARE CONFIGURATION REVIEW BOARD DISAPPROVAL REASON
SOFTWARE CONFIGURATION CONTROL BOARD MEETING DATE
SOFTWARE CONFIGURATION CONTROL BOARD DECISION
AFFECTED SPECIFICATIONS TITLE
AFFECTED SPECIFICATION NUMBER
DOCUMENT CHANGE REQUIRED (BY SPECIFICATION)
GENERATED ALTERATION NUMBERS
START DATE ENGINEERING CHANGE PROPOSAL PRODUCTION
STOP DATE ENGINEERING CHANGE PROPOSAL PRODUCTION
MANUFACTURER CODE
MANUFACTURES ITEM DRAWING NUMBER
CONFIGURATION ITEM DRAWING NUMBER
SCHEDULED ENGINEERING CHANGE PROPOSAL DELIVERY DATE
MODIFICATION TO ENGINEERING CHANGE PROPOSAL DATE
ENGINEERING CHANGE PROPOSAL MODIFYING ORGANIZATION
ENGINEERING CHANGE PROPOSAL SECURITY CLASSIFICATION
COMPUTER SOFTWARE CONFIGURATION ITEM'S AFFECTED
COMPUTER SOFTWARE CONFIGURATION ITEM'S AFFECTED
UNIT'S AFFECTED

CONFIGURATION CONTROL MANAGEMENT DATA SET

SOFTWARE PROBLEM/CHANGE REPORT DATA SET - DATA ELEMENTS

(USER DEFINED)

CONFIGURATION CONTROL MANAGEMENT DATA SET

SOFTWARE TROUBLE REPORT DATA SET - DATA ELEMENTS

SYSTEM IDENTIFICATION NUMBER
EXTERNAL TROUBLE REPORT NAME
EXTERNAL TROUBLE REPORT NUMBER
TROUBLE REPORT TYPE
ORIGINATOR'S NAME
ORIGINATOR'S ADDRESS
ORIGINATOR'S PHONE
SCREENING POINT NAME
SCREENING POINT ADDRESS
PROBLEM TYPE
NATIONAL STOCK NUMBER
TROUBLE REPORT TITLE
TROUBLE REPORT DESCRIPTION
DATE DISCOVERED
DATE LOGGED IN
STATUS
STATUS DATE
REPLY DATE
REPORT CLOSURE DATE

CONFIGURATION CONTROL MANAGEMENT DATA SET

REQUEST FOR DEVIATION/WAIVER DATA SET - DATA ELEMENTS

SYSTEM IDENTIFICATION NUMBER
TYPE (DEVIATION OR WAIVER)
DOCUMENT NUMBER
CRITICALITY
DEVIATION/WAIVER DESIGNATION
DESCRIPTION
NEED
TITLE
ORIGINATOR
ORIGINATOR ADDRESS
ORIGINATOR PHONE
IDENTIFICATION OF AUTHORIZATION
COMPUTER SOFTWARE CONFIGURATION ITEM AFFECTED
AFFECTED COMPUTER SOFTWARE CONFIGURATION ITEM VERSION
AFFECTED SPECIFICATION(S)
DATE PREPARED
DATE LOGGED
STATUS
STATUS DATE

CONFIGURATION CONTROL MANAGEMENT DATA SET

SYSTEM INFORMATION DATA SET -- DATA ELEMENTS

SYSTEM IDENTIFICATION NUMBER
SYSTEM NOMENCLATURE
SYSTEM ARMY/NAVY DESIGNATOR
LONG TITLE
PRELIMINARY DESIGN AUDIT PROJECT LEADER
PRELIMINARY DESIGN AUDIT PROJECT LEADER PHONE
PROGRAM MANAGER NAME
PROGRAM MANAGER ADDRESS
PROGRAM MANAGER PHONE
SYSTEM HARDWARE CLASSIFICATION
SYSTEM SOFTWARE CLASSIFICATION
APPLICABLE MILITARY STANDARDS
CRITICALITY NUMBER OF UNITS
CRITICALITY CHANGE RATE
CRITICALITY SIZE
CRITICALITY TRANSITION
CRITICALITY FIELDING
CRITICALITY SOFTWARE ENVIRONMENT
CRITICALITY CONFIGURATION MANAGEMENT PLAN
CRITICALITY LIBRARY STATUS
CRITICALITY CODE OWNER
CRITICALITY DOCUMENT
CONFIGURATION MANAGEMENT OFFICER
CONFIGURATION MANAGEMENT OFFICER PHONE NUMBER
CONFIGURATION MANAGEMENT OFFICER ADDRESS
CONFIGURATION MANAGER
CONFIGURATION MANAGER PHONE NUMBER
CONFIGURATION MANAGER ADDRESS
CONCEPT AND EVALUATION DATE
DEMONSTRATION AND VALIDATION DATE
RESEARCH AND DEVELOPMENT DATE
PRODUCT AND DEPLOYMENT DATE
FIELDING DATE
CONFIGURATION MANAGEMENT CENTER TRANSITION DATE
OBSOLESCENCE DATE

CONFIGURATION CONTROL MANAGEMENT DATA SET

COMPUTER SOFTWARE CONFIGURATION ITEM DATA SET - DATA ELEMENTS

SYSTEM IDENTIFICATION NUMBER
COMPUTER SOFTWARE CONFIGURATION ITEM NUMBER
DESCRIPTION OF FUNCTION
RELATED PROGRAM SPEC
VERSION IDENTIFICATION
ACTUAL VERSION NUMBER
LANGUAGE
PROGRAMMER NAME
REVISION DESCRIPTION

RELATED ENGINEERING CHANGE PROPOSAL
RELATED VERSION DESCRIPTION DOCUMENT
RELATED COMPUTER SOFTWARE CONFIGURATION ITEM
COMPLETE DATE
VERIFIED DATE
TESTED DATE
DATE VERSION UNDER CONFIGURATION MANAGEMENT

CONFIGURATION CONTROL MANAGEMENT DATA SET

SYSTEM VERSION DATA SET -- DATA ELEMENTS

SYSTEM VERSION NUMBER
ACTUAL VERSION NUMBER
VERSION TITLE
DESCRIPTION OF CHANGE
RELEASE CODE
RELATED COMPUTER SOFTWARE CONFIGURATION ITEM AND VERSION
RELATED SOFTWARE CHANGE ORDER NUMBER
APPROVAL DATE
DOCUMENTS IN REPOSITORY DATE
CODE IN REPOSITORY DATE
WORKING MASTERS DISTRIBUTED DATE

CONFIGURATION CONTROL MANAGEMENT DATA SET

SOFTWARE CHANGE ORDER DATA SET -- DATA ELEMENTS

SYSTEM IDENTIFICATION NUMBER
SOFTWARE CHANGE ORDER NUMBER
TITLE OF CHANGE
RELATED TROUBLE REPORT
RELATED CHANGE PROPOSAL
RELATED ENGINEERING CHANGE PROPOSAL (PRELIMINARY TYPE ONLY)
ORIGINATION DATE
LOGGED DATE
CONFIGURATION MANAGEMENT SPECIALIST APPROVAL DATE
SOFTWARE ENGINEER APPROVAL DATE

CONFIGURATION CONTROL MANAGEMENT DATA SET

ALTERATION DATA SET -- DATA ELEMENTS

NUMBER
MODIFICATION NUMBER
MODIFYING ORGANIZATION
MODIFICATION DATE
PARENT SOFTWARE CHANGE PROPOSAL
PARENT ENGINEERING CHANGE PROPOSAL
TITLE
ENTRY DATE
ORIGINATION DATE
ORIGINATOR
ORIGINATOR PHONE

ORIGINATOR ADDRESS
ORIGINATOR OFFICE CODE
SYSTEM IDENTIFICATION
PROGRAM IDENTIFICATION
SUBPROGRAM
MODULE NAME
MODULE NUMBER
MODULE REVISION
STATUS
STATUS DATE
RELATED ALTERATION
TAPE STATUS
KEY NAME
NEED VERIFICATION
IMPLEMENTATION DATE
VERIFICATION DATE
RESPONSIBLE PATCH PARTY
PATCH DATA
COMMENTS
PROBLEM
SOLUTION

CONFIGURATION CONTROL MANAGEMENT DATA SET

PATCH DATA SET - DATA ELEMENTS

ALTERATION NUMBER
PATCH LINE NUMBER
PATCH PAGE NUMBER
PATCH ADDRESS
OLD OCTAL
OLD INSTRUCTION
NEW PAGE NUMBER
NEW OCTAL
NEW INSTRUCTION
ASSEMBLY
COMMENTS

PDSS STANDARDS
PANEL IV
PROCEEDINGS

Panel IV meetings began with an introduction of the panel members and a review of our objective and tasks.

OBJECTIVE.

To identify changes to DOD software development standards to incorporate PDSS considerations.

Panel Tasks.

1. Determine which requirements of DOD-STD-1467 should be incorporated in current software development standards.

2. Identify changes to DOD-STD-2167 needed to incorporate PDSS considerations.

3. Identify changes to Draft DOD-STD-2168 needed to incorporate PDSS considerations.

Approach. The method of accomplishing the tasks was to have overall discussions on the PDSS environment, the status of DOD-STD-2167, suggested changes to DOD-STD-2167 and finally DOD-STD-1467. Then the panel divided into subpanels to further develop the panel reports.

DISCUSSIONS.

PDSS Activities. The workload of a PDSS activity was discussed and the functions of PDSS activities developed in Orlando I were discussed and accepted with the following additions and changes:

1. Rapid response to user software/hardware problems.

2. Problems tracking.

3. Problem analysis, including failure verification and fault isolation.

4. Problem resolution and impact analysis.

5. Development of corrections.

6. System enhancements through software changes.

* 7. Software configuration management.

* 8. Verification, validation, functional integration testing, and system level acceptance assurance testing.

9. Software production, distribution, and control.

10. Determine where and how installation of changes will be accomplished.

* 11. Software quality assurance.

12. User introduction training.

13. Software documentation maintenance.

* 14. Technical review of developing software and specifications.

* 15. Conduct PCA/FCA.

* Note: Indicates a change from Orlando I.

Responsibilities. The PDSS activity must:

1. Be responsible for investigating software and hardware problems and initiating corrective actions. A prioritization of software problems by degree of severity shall be performed. Approved software changes will be tested and verified prior to reproduction and distribution to receiving activities. These procedures will be in accordance with the information contained in the CRLCMP. Interface control documents are required to define relationships between the computer/processor system and other related systems. The PDSS activity will review and recommend approval of all changes that affect these interface areas. The responsibility of the PDSS activity extends to participation in problem solving at the interface level, and the testing of proposed solutions that impact the interface.

2. Assume responsibility for inservice engineering/logistics support of weapon system computer/processor software and related hardware.

3. Maintain and improve the software/hardware integration and test facility.

4. Provide continuing primary support to the acquisition manager, or his functional representative, and the user for assigned computer/processor software and related hardware as long as the system/subsystem remains operational, until disposal.

DOD-STD-2167 Status. Major Rick Butler gave an account of the status of DOD-STD-2167A and DOD-STD-2168. DOD-STD-2167A has been reviewed and modifications are under consideration. The major change will be to reduce the total number of DIDs by combining information. For example, DOD-STD-2167A would be modified to

have a single DID for the items that make up a product specification. This modification was of some concern to most of the members of the panel and Major Butler will relay these concerns to the review committee. Specific requirements from Section 5 of DOD-STD-2168 of will be folded into the specific requirements of DOD-STD-2167. The final version of DOD-STD-2167A will be distributed for final review in about 3 or 4 months.

PDSS Problems in DOD-STD-2167. One of the major tasks for Panel IV was to suggest PDSS related changes to DOD-STD-2167A and DOD-STD-2167. Discussions were held to list perceived problems. No attempt was made to limit the ideas for consideration. The following issues were raised:

1. Does not contain a strong pass down requirement.

2. Does not contain a strong traceability requirement.

3. Does not adequately address final preparation for delivery.

4. Does not adequately address the program build process.

5. Does not accommodate modification to existing documents.

6. There is no stress testing requirement.

7. Does not address degree of rigor required for software Quality Assurance during PDSS.

8. There is no definition of Preliminary Software Development Activities.

9. There is no definition of Post Software Development Activities.

Other Issues Raised but not Pursued.

1. Funding.

2. PDSS Contract procurement.

3. Firmware resolution.

Another panel task was to identify which of the requirements of DOD-STD-1467 should be incorporated into DOD-STD-2167. Mr. Chuck Gordon (CACI, Inc.) gave a briefing on DOD-STD-1467.

Subpanel Reports. Panel IV was then divided into the following subpanels.

Subpanel A. Determine which requirements of DOD-STD-1467 should be incorporated in current software development standards.

Thomas Conrad
Jim Heil (Subpanel Chairperson)
Kurt Krabbee
Dan Kvenvold

Subpanel B. Identify changes to DOD-STD-2167 needed to incorporate PDSS considerations by analyzing DOD-STD-2167 to identify items that don't support PDSS.

Greg Bornako
Paul Byerley
Jim Parlier
Jane Radatz (Subpanel Chairperson)
Jack Reichson
Wayne Sherer
James Steenwerth
Mae Stees

Subpanel C. Identify changes to DOD-STD-2167 needed to incorporate PDSS considerations by analyzing the standard against the identified PDSS problems.

Karen Bausman
Rick Butler (Subpanel Chairperson)
David Castellano
Ole Golubjatnikov
Charles Kelly
David Maibor
Lee Stewart

The three subpanels reviewed the suggested changes to the software development standards. The entire panel recommends the following changes be made to the standards:

1. Describe the post deployment phase.

2. Define the preliminary software development activities.

3. Address modification to other than DOD-STD-2167 developed items within a DOD-STD-2167 environment.

4. Change title to: "Defense Systems Software Development and Support."

5. Incorporate identified items from DOD-STD-1467 into DOD-STD-2167.

6. Incorporate identified items from DOD-STD-1467 DIDs into DOD-STD-2167 DIDs.

7. Incorporate changes identified by subpanel reviews into DOD-STD-2167.

8. Incorporate changes to emphasize the software build process.

9. Add transition information to the CRISD DID.

10. Provide a means for delivery of documentation for commercially available software in DOD-STD-2167.

Methods. The following methods could be used to incorporate these items into the software development standards.

1. Modify DOD-STD-2167 in the following ways:

(a) Add an appendix to give top level guidance, provide the same information as their body of the standard except from a PDSS perspective or add an appendix to explain how to modify paragraphs in the body of the standard for PDSS.

(b) Rewrite paragraphs in the body of the standard by modifying existing paragraphs for PDSS or by adding shadow paragraphs.

2. Develop a parallel PDSS standard.

3. Develop a PDSS handbook.

The panel preferred option 1, modify the existing DOD-STD-2167. Two panel members felt that a separate PDSS standard was needed.

Recommendation. The JLC should review the panels recommendations for inclusion into the software development standards.

Subpanel Reports. The attached are the reports generated by Panel IV subpanels.

Subpanel A. Determine which requirements of DOD-STD-1467 should be incorporated in current software development standards.

Subpanel B. Determine the applicability of DOD-STD-2167A to a PDSS environment.

Subpanel C. Determine the applicability of the issues raised and general PDSS concerns affecting DOD-STD 2167.

PANEL IV
SUBPANEL A REPORT

OBJECTIVE.

Determine which requirements of DOD-STD-1467 should be incorporated in current software development standards.

Approach. To address the objective, Subpanel A reviewed DOD-STD-1467 and the four related DIDs to determine which items needed to be mapped into DOD-STD-2167 and its related DIDs. Issues IV and V are the results of that review. In addition, several general recommendations were made and are found in Issues I, II and III. The May, 1985 Joint Regulation was reviewed and the results are found in Issue VI.

ISSUES. The following six issues were addressed:

Issue I. Position on DOD-STD-2167 (Relevant to PDSS).

It was unanimously agreed (by Subpanel A) that DOD-STD-2167 should address PDSS concerns:

1. During initial development (prior to deployment) to facilitate successful post development activities/operations.

2. For use during the post deployment phase (for software changes, etc.)

Issue II. The need for a definition of when post deployment starts was recognized. Although no one definition is adequate for all cases, the following nominal definitions were recommended for this purpose:

1. Post deployment starts when the system has successfully completed the installation and checkout at a site/location/platform and is available for operational use. If dealing with multiple sites, separate installation dates occur at each site, but the post deployment phase starts on the the first site's installation/deployment date. This will cause some overlap of the post deployment phase of the installation/deployment activities for other sites, until all sites have been deployed.

2. DOD-STD-2167 does not identify a post deployment phase. Note that Figure 4, in Appendix B of DOD-STD-2167, shows only a 'PRODUCTION & DEPLOYMENT PHASE', ending in 'SYSTEM RETIREMENT'. Recommend that the figure be modified as shown in Figure 4 of this document, and that related text (e.g.; paragraph 20.4.4) be added to clearly show a 'PRODUCTION & DEPLOYMENT PHASE' and a missing 'POST DEPLOYMENT PHASE'.

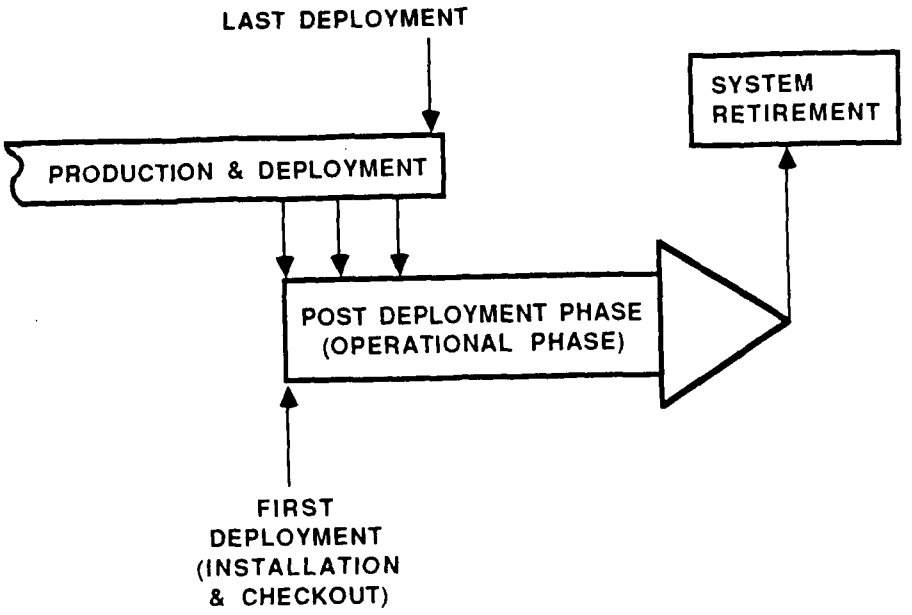


FIGURE 4. Suggested Modifications to Figure 4, DOD-STD-2167, "Sample System Support Cycle".

Issue III. DOD-STD-2167 needs some minor improvements to fully address PDSS considerations.

Recommendation 4-4-01. Add two Appendices to DOD-STD-2167A to:

1. Describe the PDSS process (similar to Appendix B)

2. Provide guidelines for all three PDSS change categories:

   - o Major changes.
   - o Routine (minor) changes.
   - o Emergency/urgent changes.

Advantages. Handles PDSS concerns in ways that are:

1. Modular for PDSS users (overview/roadmap).

2. Easy to incorporate into DOD-STD-2167A.

Issue IV. There are some items in DOD-STD-1467 that should be added into DOD-STD-2167A and the related DIDs.

Recommendation 4-4-02. Add into DOD-STD-2167A the items numbered 1 through 17 in the attached copy of DOD-STD-1467. Table 2 lists

these items and provides amplifying information. Add into DOD-STD-2167A the recommendations derived by reviewing the DIDs from DOD-STD-1467. The results of the DID review follows:

DID DI-E-7141. Documentation of Commercially Available or Privately Developed Software.

Discussion. The documentation requirements in DOD-STD-2167A do require the contractor to provide such data in paragraphs 4.4, 4.6, 4.7, 4.9, 4.10, 5.1.1.3e, 5.2.1.9, 5.2.1.10, 5.2, and other paragraphs in the Plans and Manuals DID.

Recommendation 4-4-03. Paragraph 4.4 of DOD-STD-2167A refers to commercially available, reusable and GFS software. "Reusable" generally can be interpreted to mean "previously developed," but, for clarity, it is recommended the term, "previously developed," be added to the paragraph title. It is also recommended that the second sentence of paragraph 10.2 of DI-E-7141, minus the word "evaluation", be added to appropriate paragraphs in DOD-STD-2167A DIDs relative to commercially available documentation requirements.

DID DI-E-7142. Software Support Transition Plan.

Discussion. DID DI-E-7142 is appropriate only for the case when the life cycle software support environment is being procured from the developer of the operational software to be supported with that environment. It is not useful in the case where the government already owns an environment intended for the post deployment support of the acquired operational software, or in the case where the post deployment support environment and operational software are procured from different vendors. Most of the useful data required by this DID is found in the CRLCMP.

Recommendations:

1. Recommendation 4-4-04. Modify the CRISD (DI-MCCR-80024) DID to incorporate a section on transition management similar to the CRLCMP discussion on the subject.

2. Recommendation 4-4-05. Modify DOD-STD-2167A to require a post deployment supportability demonstration prior to software product acceptance, rather than rely on an additional DID for an environment transition plan. The supportability demonstration would include, as a minimum, complete regeneration from operational software source using only delivered or government owned software, hardware and documentation.

3. Recommendation 4-4-06. Modify DOD-STD-2167A to include a PDSS Readiness Review at which the post deployment supportability demonstration is to be conducted. In addition, MIL-STD-1521 may need to be modified to reflect these changes.

Table 2 - RECOMMENDED CHANGES TO JOINT REGULATION

Paragraph #	Recommended Change
2.2	...Production & Deployment... d. " & " Need support during it's operational life. Figure 3 - needs delta change for Post Deployment.
2.5.2	CRLCMP - emphasize the post deployment support.
2.5.4	CRWG - okay.
2.5.12	Include risks related to PDSS.
2.5.15	FW - not consistent with Fall T.A.W.G recommendations. changes - see Austin Maher.
2.6.1	Add draft RFP to bidders to get comments related to PDSS and Tailoring.
2.6.2.6	Okay.
2.6.3	Add PDSS considerations.
2.7.3.1	Add PDSS and software support considerations.
2.7.3.2	SQEP, SQPP.
5.2.1.5.3	Item d.- add: building, including rebuilding and reloading the software.
5.2.1.6.4	Okay "but not final" - support planning.
5.2.3	Add PDSS considerations.
5.2.4	Update support documents - DOD CRLCMP.
5.4.7	FQR - add PDSS staff (also, reflect FQR in DOD-STD-2167, DOD-STD-2168).
2.2.1.4	Okay, but beef up (reflect some - STD) b. Add "support software".

(Intentionally Blank)

DID DI-E-7143. Life Cycle Software Support Environment Users Guide.

Discussion. The requisite information delineated in DI-E-7143 is adequately described in DOD-STD-2167A DIDs; DI-MCCR-80018 through DI-MCCR-80024. The majority is covered in the CRISD, with the exception of software performance evaluation.

Recommendation 4-4-07. Include paragraph 10.4.4.4.1, Software Performance Evaluation, of DI-E-7143, or something very similar, in the CRISD.

Issue V. It is cost effective to procure the PDSS support environment as a derivative of the developmental support environment. DOD-STD-1467(AR) requires the contractor to identify all variances between the developmental environment actually utilized and the PDSS support environment to be delivered. This information is useful to the procuring agency to ensure an adequate PDSS support environment. DOD-STD-2167 does not clearly require the equivalent data.

Recommendations:

1. Recommendation 4-4-08. DOD-STD-2167 be applied to the delivery of the PDSS support environment.

2. Recommendation 4-4-09. Incorporate the environment variance analysis of DOD-STD-1467 into the CRISD of DOD-STD-2167.

3. Recommendation 4-4-10. Incorporate other aspects of DOD-STD-1467 into DOD-STD-2167 as identified in the other recommendations. [See Table 3 and DOD-STD-1467, included in proceedings]

ISSUE VI. The Joint Regulation for PDSS issues, "Management of Computer Resources in Defense Systems (Draft)" - Old Version, dated 7 May 1985, should be modified to reflect current PDSS requirements.

Recommendation 4-4-11. Modify the Joint Regulation as defined in Table 3.

(Intentionally Blank)

Table 3 - DOD-STD-1467 ITEMS TO INCORPORATE INTO DOD-STD-2167

Item #	1467 Section	Comments
1	1.1 (partial)	Expand the purpose of DOD-STD-2167 to include the essence of...
2		Definitions (2a-2h) included in DOD-STD-2167
2a	3.3	Reword.
2b	3.5	
2c	3.7, 3.7.1, 3.7.2	
2d	3.9.1	
2e	3.9.2	
2f	3.9.3	"GFS".
2g	3.9.4	Reword.
2h	3.10	
3	4.1	Address build function in DOD-STD-2167, paragraph 5.3.1.17.
4	5.1.2 (partial)	For Standard
5	5.1.4	Cover software build (eg; link/load) in SDP (paragraph 10.2.5.2.3) & in Programmers Manual (para 10.2.3.3).
6	5.1.5 (partial)	Compare to CRISD.
7	5.1.5 (partial)	
8	5.2.1.2	Check Standard, CRISD, CM Plan & SDP.
9	5.2.2 (partial) 5.2.3 (partial)	Include in Standard.
10	5.2.2.1 (partial)	Include in Standard. Address in CRISD for support activity.

Table 3 - DOD-STD-1467 ITEMS TO INCORPORATE INTO DOD-STD-2167 (CONT'D)

Item #	1467 Section	Comments
11	5.2.2.5 (partial) 5.2.3.5 (partial)	Include in Standard.
12	5.2.3.1 (partial)	Include in Standard.
13	5.3 - 5.3.4	Evaluate for general inclusion in the Standard and the CRISD.
14	5.3	Demonstrate software support environment at contractor facility & PDSS Activity facility in the Standard.
15	5.3.1 (partial)	Verify in CRISD.
16	5.3.2.(partial)	Include in Standard.
17		Add to CRISD recommendation on transition from DCSSE to PDSS Activity of deliverable software.

PANEL IV
SUBPANEL B REPORT

OBJECTIVE.

The objective of Subpanel B was to determine the applicability of DOD-STD-2167A to a PDSS environment.

Approach. Subpanel B considered each of the detailed requirements in Sections 5.1 through 5.7 of DOD-STD-2167A to determine their applicability to a PDSS environment. Table 4 shows the results of that analysis. The overall findings were that:

1. Nearly all requirements of DOD-STD-2167A apply to PDSS but need rewording to accommodate an environment of change rather than an initial development environment. The following general problems in DOD-STD-2167A were identified:

a. No testing baseline.

b. "Disapproval" does not work for previously baselined documents.

c. Software that needs to be changed is often not a CSCI.

d. Subpanel B wanted system integration and test planning and performance left in.

2. A few new requirements were identified for the Software Requirements Analysis phase in a PDSS environment. These are:

a. Physical configuration evaluation

b. Review existing documentation

c. Updates to CRISD

Options. Subpanel B also considered six options for implementing its findings. These options are listed below. A trade off study of these options (methods to enhance DOD-STD-2167A) is presented in Table 4.

1. Writing a parallel stand alone standard for PDSS.

2. Rewriting DOD-STD-2167A to include a related PDSS paragraph below each development paragraph.

3. Rewriting DOD-STD-2167A to include a PDSS version as an appendix to DOD-STD-2167A.

4. Rewording each existing DOD-STD-2167A paragraph so that each paragraph applies to either development or PDSS.

5. Include guidance for rewording each paragraph in an appendix to DOD-STD-2167A.

6. Include guidance for rewording each paragraph in a handbook.

Resulting Issue. The concept and approach of DOD-STD-2167A apply to a PDSS environment, but the specific wording of most paragraphs implies an environment of new development rather than an environment of change.

Recommendations. Subpanel B recommends the following actions be taken:

1. Recommendation 4-4-12. Develop a specific rewrite or mandatory modifications to each paragraph of DOD-STD-2167A for a PDSS environment.

2. Recommendation 4-4-13. Perform a trade-off study to determine whether to include the results of step 1 in an appendix to DOD-STD-2167A, some other method of modification to DOD-STD-2167A, develop a separate PDSS standard, or develop a PDSS handbook. (See options identified in the approach above).

3. Recommendation 4-4-14. Include specific wording for each additional task required during PDSS. (See Table 5).

4. Recommendation 4-4-15. Use the findings in Tables 4 and 5 as the basis for these activities.

Table 4. TRADE-OFF STUDY FOR METHODS TO ENHANCE DOD-STD-2167A

Method 1. Write a parallel stand alone standard for PDSS.

Benefits.

- o Clear distinction between PDSS and development phases.
- o Eliminates confusion.
- o Less tailoring required.
- o Easier to write RFP.
- o No impact on DOD-STD-2167A publication date.

Detractors.

- o Hard to keep synchronized with other standards.
- o Additional administrative overhead.
- o Time to develop a new standard.
- o Violates streamlining initiative.

Method 2. Rewrite DOD-STD-2167A to include a related PDSS paragraph below each development paragraph.

Benefits.

- o One standard for whole life cycle.
- o Eases standards maintenance.
- o Keeps ACQ/PDSS in sync.
- o Can do more referencing than under separate cover.

Detractors.

- o Standard becomes twice as thick.
- o Makes standards coordination more difficult (2 communities).
- o Duplication within standard.

Table 4. TRADE-OFF STUDY FOR METHODS TO ENHANCE DOD-STD-2167A (Cont'd)

Method 3. Rewrite to include a PDSS version as an appendix to DOD-STD-2167A.

Benefits.

- o One standard for whole life cycle.
- o Eases standards maintenance.
- o Keeps ACQ/PDSS in sync.
- o Less confusing than side by side presentation.
- o Can do more referencing than under separate cover.

Detractors.

- o Not as easy to spot ACQ/PDSS differences.
- o Standard becomes twice as thick.
- o Makes standards coordination more difficult (2 communities).
- o Duplication within standard.

Method 4. Reword each existing DOD-STD-2167A paragraph so that each paragraph applies to either development or PDSS.

Benefits.

- o One standard for whole life cycle.
- o Eases standards maintenance.
- o Keeps ACQ/PDSS in sync.

Detractors.

- o Language is convoluted and highly confusing.
- o Slows down approval if attempted in Revision A.
- o Makes tailoring more difficult.
- o Potential impact on DOD-STD-1467 (redundancy).

Table 4. TRADE-OFF STUDY FOR METHODS TO ENHANCE DOD-STD-2167A (Cont'd)

Method 5. Include paragraph by paragraph modification instructions for PDSS in an appendix.

Benefits.

- o No synchronization problem.
- o One standard for whole life cycle.
- o Keeps ACQ/PDSS in sync.
- o Shorter than two full sets of requirements.
- o No duplication within standard.

Detractors.

- o Not clear this should exist in contractor document.
- o Harder to use than completely written PDSS version.
- o Tailoring on top of tailoring is needed.

Method 6. Include guidance for rewording each paragraph in a handbook.

Benefits.

- o Tailoring guidance is in a document that addresses the correct audience.
- o Gives formal help to PMs for applying the standard to PDSS.

Detractors.

- o A handbook is not a standard.
- o Requires program manager to apply all these requirements by SOW.
- o Less visible than an appendix in a standard.
- o Raises question of a similar approach for other communities.

(Intentionally Blank)

Table 5. DOD-STD-2167A PDSS ANALYSIS

APPLICABILITY TO PDSS

SECTION	YES	NO	COMMENTS
5.1.1.1	x		
5.1.1.2	x		But SSS may need to be updated.
5.1.1.3 a	x		May be tailored out.
5.1.1.3 b	x		
5.1.1.3 c		x	Contracting agency provides.
5.1.1.3 d		x	Contracting agency provides.
5.1.1.3 e	x		
5.1.1.3 f	x		
5.1.1.3 g	x		Not strong enough.
5.1.1.3 h		x	
5.1.1.3 i	x		
5.1.1.4	x		
5.1.1.5		x	
5.1.1.6		x	
5.1.1.7	x		
5.1.1.8	x		Delete "preliminary".
5.1.1.9	x		Modify for existing CSCI. Intent OK.
5.1.1.10	x		Last sentence needs to be separated for tailorability.
5.1.1.11	x		Contractor detected problems.
5.1.2	x		

Table 5. DOD-STD-2167A PDSS ANALYSIS (Cont'd)

APPLICABILITY TO PDSS

SECTION	YES	NO	COMMENTS
5.1.2.1	x		
5.1.2.1		x	
5.1.2.3		x	
5.1.2.4		x	
5.1.2.5	x		Modify flavor to PDSS.
5.1.3	x		Modify for PDSS.
5.1.4	x		Needs rewording.
5.2.1.1	x		
5.2.1.2	x		But modified flavor.
5.2.1.2	x		But modified flavor.
5.2.1.4	x		Use existing design technique.
5.2.1.5	x		Develop and modify.
5.2.1.6	x		Develop and/or modify test plans as applicable.
5.2.1.6.a	x		
5.2.1.6.b	x		
5.2.1.6.c	x		
5.2.1.6.d	x		
5.2.1.6.e	x		Unclear wording.
5.2.1.7	x		Delete first sentence. Step 1 should be under Government control. Prepare revisions, and submit changes. Disapproved a problem.

Table 5. DOD-STD-2167A PDSS ANALYSIS (Cont'd)

APPLICABILITY TO PDSS

SECTION	YES	NO	COMMENTS
5.2.1.8	x		Develop and/or modify. Keep bridge to system testing.
5.2.1.9	x		(Update, not write) Army does not use OM.
5.1.2.10	x		Update not written.
5.1.2.11	x		Update not written.
5.2.1.12	x		Add to previous plan as well.
5.2.1.13	x		
5.2.2.1	x		OK as is, and absolutely necessary.
5.2.2.2	x		Modify not produce new.
5.2.2.3	x		Modify not produce new.
5.2.2.4	x		Modify not produce new
5.2.2.5	x		Modify not produce new
5.2.2.6	x		Modify not produce new
5.2.3	x		Review wording for PDSS flavor. "If required" needed. (Preliminary design may have to be modified).
(General note: Make sure we have the flavor of analyzing change requests to see what needs to be changed).			
5.2.4	x		No developmental configuration. This is product baseline being changed.
(General note: Is a major modification or redevelopment performed in accordance with current DOD-STD-2167A or modified as we are discussing here?).			
5.3.1.1	x		Reflavor for PDSS

Table 5. DOD-STD-2167A PDSS ANALYSIS (Cont'd)

APPLICABILITY TO PDSS

SECTION	YES	NO	COMMENTS
5.3.1.2	x		Reflavor for PDSS.
5.3.1.3	x		
5.3.1.4	x		May want to require retaining existing design methods.
5.3.1.5	x		
5.3.1.6	x		May get original ones but it's unlikely. Leave "establish." These are new SDFs.
5.3.1.7	x		Updates & new.
5.3.1.8	x		Reflavor.
5.3.1.9	x		Reflavor.
5.3.1.10	x		Reflavor (keep regression testing invisible here).
5.3.1.11	x		Reflavor.
5.3.1.12	x		Reflavor.
5.3.1.13	x		Not internal control. Disapproval is wrong.
5.3.1.14	x		Add regression testing.
5.3.1.15	x		OK as is.
5.3.1.16	x		"update" or OK as is.
5.3.1.17	x		
5.3.1.18	x		Needed only if there is a hardware change.
5.3.1.19	x		
5.3.2	x		Product paragraphs follow comments for corresponding activity paragraphs.

Table 5. DOD-STD-2167A PDSS ANALYSIS (Cont'd)

APPLICABILITY TO PDSS

SECTION	YES	NO	COMMENTS
5.3.3	x		Reflavor.
5.3.4	x		As in 5.2.4 product baselines. Contractor controls change pages as contractor development configuration.
5.4.1.1	x		Reflavor.
5.4.1.2	x		Reflavor. May not have SDP, SSPM.
5.4.1.3	x		There will usually be coding standards.
5.4.1.4	x		Needs improvement for both development and PDSS environment. Require use of PDSS owned equipment.
5.4.1.5	x		Reflavor.
5.4.1.6	x		Reflavor. Test new or changed units.
5.4.1.7	x		
5.4.1.8	x		Design document is updated by submitting change request.
5.4.1.9	x		
5.4.1.10	x		Update and develop as necessary.
5.4.1.11	x		Update as required.
5.4.1.12	x		Earlier notification needed.
5.4.1.13	x		Refocus. Add regression testing.
5.4.1.14	x		OK as is.

Table 5. DOD-STD-2167A PDSS ANALYSIS (Cont'd)

APPLICABILITY TO PDSS

SECTION	YES	NO	COMMENTS
5.4.1.15	x		
5.4.2	x		As above for 5.3.2.
5.4.3	x		Development configuration only the appropriate portions.
5.5.1.1	x		For formal baselines: Contractor follows Government CM Plan. Avoid tone of "go ahead then be told later that it is not OK."
5.5.1.2	x		As required. (May involve unchanged units).
5.5.1.3	x		Be careful about the handling of formal baselines.
5.5.1.4	x		As necessary.
5.5.1.5	x		
5.5.1.6	x		Limit to development configuration.
5.5.1.7	x		As necessary. Refer back to original SPR.
5.5.1.8	x		Applies tone/modified procedures only.
5.5.1.9	x		
5.5.1.10	x		
5.5.1.11	x		
5.5.2	x		As above in 5.3.2.

Table 5. DOD-STD-2167A PDSS ANALYSIS (Cont'd)

APPLICABILITY TO PDSS

SECTION	YES	NO	COMMENTS
5.5.3	x		
5.5.4	x		See earlier discussion.
5.6.1.1	x		Same as 5.5.1.1.
5.6.1.2	x		
5.6.1.3	x		Concern about "independent words".
5.6.1.4	x		In handbook, suggest high frequency status reports.
In STP DID, emphasize need to state test environment and its relationship to the Government PDSS environment.			
5.6.1.5	x		Reflavor.
5.6.1.6	x		See 5.5.1.6.
5.6.1.7	x		Prepare proposed new VDD of proposed revision to existing VDD.
5.6.1.8	x		Complete revisions.
5.6.1.9	x		
5.6.2	x		As above.
5.6.3	x		Review words for PDSS.
5.6.4	x		Review words for PDSS.
5.6.5	x		
5.6.6	x		
5.7	x		Include regression testing.
5.7.1.1	x		Rewording needed.
5.7.1.2	x		"Independent" is a concern.
5.7.1.3	x		

Table 5. DOD-STD-2167A PDSS ANALYSIS (Cont'd)

APPLICABILITY TO PDSS

SECTION	YES	NO	COMMENTS
5.7.1.4	x		Might revise higher development documents.
5.7.1.5	x		But is unclear.
5.7.2		x	Not applicable.

PANEL IV
SUBPANEL C REPORT

OBJECTIVE.

To examine the applicability of specific PDSS issues and general concerns affecting DOD-STD-2167.

ISSUES.

The following issues were broken into DOD-STD-2167 areas, and action is proposed where appropriate (Recommendation 4-4-16).

Issue 1. DOD-STD-2167 does not contain a strong pass-down requirement to subcontractors to conform to DOD-STD-2167.

a. Discussion. The pass-down requirements are covered on page 16 of DOD-STD-2167, paragraphs 4.5, 5.1.1.h and 5.8.1.6; page 24 of DOD-STD-2168, paragraph 4.11; and page 20 of Software Development Plan DID, paragraph 10.2.7.3.2.1.8.

b. Action. Strike second sentence in paragraph 4.5 of DOD-STD-2167, to make the paragraph apply to all software delivered/procured from subcontractors.

Follow-up the pass down requirements when specific actions from DOD-STD-2167 are consolidated in DOD-STD-2167A rewrite.

Issue 2. DOD-STD-2167 does not have a strong traceability requirement to lower specifications.

a. Discussion. Upon examination, the following documents were found to be weak in traceability requirements:

1. SSS DI-CMAN-80008 paragraph 10.2.5.1.6.1.2.

2. SRS DI-MCCR-80025, paragraph 10.2.5, 10.2.5.8, Table IX.

3. STLDD DI-MCCR-80012, paragraph 10.2.5.2, Table I.

4. SDDD DI-MCCR-80031, paragraph 10.2.5.3.1.1, Table V, 5.1.1.5, 5.1.1.6, 5.2.1.2, 5.3.1.2.

5. DOD-STD-2167, paragraph 5.8.1.2.3b, 5.8.1.2.4a, 5.8.1.2.5b, 5.8.1.2.6c, 5.8.1.2.7b, 5.8.1.2.8c.

b. Action. Track the consolidation to ensure the traceability areas are maintained.

Issue 3. DOD-STD-2167 does not fully address final preparation and delivery of the software.

a. Discussion. The standard, and data items, which address delivery are as follows:

1. DOD-STD-2167, paragraph 5.6.2.4

2. DI-MCCR-80025, paragraph 10.2.7.

b. Action. Correct paragraph 5.6.2.4 to have preparation and delivery IAW the System Requirement Specification (SRS) or specified governing document (Program Identification Document (PID), Configuration Identification Document (CID), SOW).

Issue 4. DOD-STD-2167 does not fully address the software build process.

a. Discussion. There are multiple definitions to the term "build" for software. Paragraph 4.1.2 of DOD-STD-2167 addresses the iteration process. The ability to allow the contractor to define the software development process in the SDP in regards to iterative/builds is proposed.

b. Action. Add paragraph 4.1.2 to DOD-STD-2167, requiring that the contractor define how the iterative or build process is to work. The description should go into the SDP.

Issue 5. DOD-STD-2167 does not address the modification of non-DOD-STD-2167 DIDs within a DOD-STD-2167 development philosophy.

a. Discussion. The ability to apply the standard, independently of the DIDs, needs to be resolved for PDSS. This would allow the development philosophy to be required while permitting the previous standards to be revised.

b. Action. The JLC should review how to avoid costly document conversion to DOD-STD-2167 DIDs format, investigate and provide methods, acceptable to OSD, to invoke DOD-STD-2167 requirements during any life cycle phase (e.g. PDSS). This would allow contractor and government support agency to update any existing non-DOD-STD-2167 documentation. Include guidance on cost/benefit trade-off factors.

Issue 6. What is the degree of rigor the software quality assurance has during PDSS.

a. Discussion. There is a current lack of software quality evaluations that PDSS activities could track for the various support functions.

b. Action. The consolidation of DOD-STD-2167 and DOD-STD-2168 into a functional standard will address the issue. This folding-in of requirements requires tracking to ensure quality checks are supporting the activities that PDSS addresses.

Issue 7. There is no definition of preliminary software development activities.

a. Discussion. DOD-STD-2167 currently starts the software development phase assuming a System Segment Specification (SSS) has been initially completed. The area of preliminary software development would involve the SSS and requires a review.

b. Action. Until MIL-STD-499 is updated, DOD-STD-2167 should develop the pre-software development requirements and add the tasks as an appendix. When MIL-STD-499 is updated the appendix can be removed.

Issue 8. There is no definition of Post Deployment activities for software in DOD-STD-2167. Is there a need for a separate PDSS standard.

a. Discussion. There is no real consensus that a separate standard is needed. However, a two page PDSS activities description could help. In addition, Figure 2 and CRISD DID DI-MCCR-80024 should be updated to reflect PDSS planning. Guidance for the program officers is lacking on PDSS application.

b. Action.

1. Update Figure 2 of DOD-STD-2167 to illustrate PDSS phase.

2. Produce a PDSS activities description for DOD-STD-2167 that defines specific actions and how and where to re-enter the DOD-STD-2167 cycle to affect the required change/ support (see attached chart).

3. Review the CRISD DID, DI-MCCR-80024 to enhance it for a living PDSS plan that allows for the transitioning of the contractors knowledge.

4. Provide guidance to the program office for including PDSS actions within the proposed DOD-HDBK-287 that supports the requirements of DOD-STD-2167.

Issue 9. Deliverable data under DOD-STD-2167 and the interaction with MIL-S-83490 is misunderstood.

a. Discussion. The ability to tailor DOD-STD-2167 DIDs under MIL-S-83490 is not addressed. MIL-S-83490 allows format control of DIDs from total compliance to commercial practices for DID contents.

b. Action. The tailoring guidance would address that MIL-S-83490 may be applied to DOD-STD-2167 DIDs to allow flexibility in receiving previously developed or existing documentation standards for specifications, plans, manuals and reports.

Issue 10. The title of DOD-STD-2167 does not accurately reflect the scope of the standard.

a. Discussion. The standard may be applied during any Life Cycle phase. PDSS reflects a specific phase that the standard is applicable to.

b. Action. Recommend that the title state the support aspect of DOD-STD-2167 e.g., Defense System Software Development and Support Standard.

PANEL IV

WORKING DOCUMENT

(DOD-STD-1467)

The following document contains the penned, marginal notations that Panel IV made to reference the various points that were discussed. This is included to provided additional insight into several of those points.

(Intentionally Blank)

DOD-STD-1467 (AR)
18 JANUARY 1985

MILITARY STANDARD SOFTWARE SUPPORT ENVIRONMENT

AMSC A3432

AREA ECRS

(Intentionally Blank)

DEPARTMENT OF DEFENSE
Washington, DC 20301

Software Support Environment.

DOD-STD-1467 (AR)

1. This Military Standard is approved for use by the U. S. Army Armament, Munitions and Chemical Command (AMCCOM), Department of the Army, and is available for use by all Departments and Agencies of the Department of Defense.

2. Beneficial comments (recommendations, additions, deletions) and any pertinent data which may be of use in improving this document should be addressed to: Commander, AMCCOM, Attn: SMCAR-TSB, Dover, NJ 07801 by using the self-addressed Standardization Document Improvement Proposal (DD Form 1426) appearing at the end of this document, or by letter.

(Intentionally Blank)

FOREWORD

1. This Standard defines the efforts necessary to ensure the existence of a complete life-cycle software support capability for the contractually deliverable software when it enters the operational inventory. During the operation and maintenance phase of the life cycle, a designated life cycle software support activity will be responsible for managing the contractually deliverable software and for ensuring that any changes are controlled and coordinated with other functional areas. In most cases, an existing life cycle software support activity will be assigned the added responsibility to support the new contractually deliverable software. The added work-load for the new software must be integrated into the existing life cycle software support activity. Only under unusual circumstances will a new facility, or significant additions to existing facilities, be possible.

2. Life cycle software support activities generally support their assigned responsibilities through a support system and a standard set of software, integrated with equipment and unique software for each target computer system. The support systems and software are also standardized to a certain extent among the different life cycle software support activities. For each target computer system, the designated life cycle software support activity will ultimately be responsible for a complex mix of existing standard, newly developed, commercially available and privately developed software for both operational and support functions. The objective of this Standard is to ensure that the contractually deliverable software will be supportable by the designated life cycle software support activity throughout the software's life cycle.

3. This Standard is designed to recognize the needs and constraints of existing life cycle software support activities and, at the same time, allow the software contractor the flexibility to develop software and manage the contract in accordance with the contractor's best judgement and practices. Accordingly, this Standard does not dictate the approach to be used by the contractor. The contracting activity will normally identify in the request for proposal, the designated life cycle software support activity and any of its items that are designated for use by the contractor. Subject to the constraints imposed by the contracting activity, the contractor may propose to use the existing resources of the contracting activity, to use the contractor's own resources (either existing or to be developed), or to select from a wide range of options in between. The contractor will identify the selected approach in the proposal for the contracted software effort. The contractor's approach will be considered during source selection and is subject to approval by the contracting activity prior to implementation.

(Intentionally Blank)

CONTENTS

(Intentionally Blank)

(Intentionally Blank)

(Intentionally Blank)

1. SCOPE

1.1 Purpose. This Standard establishes uniform minimum requirements for the contractor to define a Developmental Software Support Environment (DSSE), to ensure the compatibility of this environment with a contracting activity's designated Life Cycle Software Support Environment (LCSSE), and to ensure the existence of a complete contracting activity life cycle software support capability for the deliverable software of the contracted effort. (1)

1.2 Application. When invoked in a statement of work, these requirements shall apply to all software and associated items necessary to develop and support the software that is deliverable under the contract.

1.3 Contractual intent. This Standard is intended to be augmented by the contracting activity in statements of work in order to satisfy particular development and support requirements for each contracted software effort. The prime contractor is responsible for invoking all requirements of this Standard on any and all subcontractors, vendors or other sources involved in the development of software to be delivered under the requirements of the contract. The prime contractor is responsible for ensuring that all subcontractors, vendors, or other sources involved in the delivery of software to be used to fulfill the requirements under the contract, comply with the requirements of this Standard.

(1)

(Intentionally Blank)

2. REFERENCED DOCUMENTS

2.1 Issues of documents. None.

(Intentionally Blank)

3. DEFINITIONS

3.1 Introduction. The definitions provided in this Section describe the terms as they are used in this Standard.

3.2 Contracting activity. The contracting activity refers to that office, with contract and project directive administrative authority, which has prime responsibility for and authority over the contracted software effort.

3.3 Contracting activity designated resources. Resources that the contracting activity identifies to the contractor to be included and used in the Developmental Software Support Environment. (Note: The contracting activity may elect to furnish these resources and arrange any necessary licenses, or require the contractor to do so).

3.4 Contractor. Contractor refers to any organization under contract or tasking agreement with the contracting activity to perform any part of the contracted software effort.

3.5 Host computer system. Computer equipment, support software, or procedures used to develop and support the contractually deliverable software for one or more target computer systems. A host computer system may additionally include: a) elements of the target computer systems, b) modifications, emulations, or simulations of the target computer systems, or c) specially designed software or equipment to permit development and support of the operational and support software.

3.6 Previously developed. Software and documentation that is available for delivery and acceptance prior to award of the contract.

3.7 Software support environment. A host computer system, plus other related equipment and procedures, located in a facility that provides a total support capability for the software of a target computer system (or a set of functionally and physically related target computer systems). The environment enables the performance of a full range of services, including: performance evaluation, system and software generation, development and testing of changes, simulation, emulation, training, software integration, configuration management, and operational distribution for the software. Two types of software support environments are addressed:

3.7.1 DSSE (Developmental Software Support Environment). Those contracting activity approved resources identified by a software contractor to be used to support the software requirements under the contracted efforts.

3.7.2 LCSSE (Life Cycle Software Support Environment). Those contracting activity resources used by the life cycle software support activity to provide a total life cycle software support capability for assigned target computer systems.

(Intentionally Blank)

3.8 Software. A combination of associated computer programs and computer program data definitions required to enable the computer hardware to perform computational or control functions. (Note: this definition includes firmware within its applicability. This definition of software is independent of the type of physical storage media in which the software resides). Software is further defined as follows:

3.8.1 Operational software. All software used to operate, or that is resident in, a target computer system.

3.8.2 Support software. All software used to aid the development, testing and support of operational software. Support software includes all software used to operate, or that is resident in, a software support environment.

3.9 Software sources. For the purposes of this Standard, the following terms are used to describe the sources of software:

3.9.1 Commercially available software. Previously developed software used regularly for other than Government purposes and sold, licensed or leased in significant quantities to the general public at established market or catalog prices.

3.9.2 Privately developed software. Previously developed software independently developed by an industrial source at its own expense. (Note: In contrast with software that is commercially available, this software may have limited availability and may be subject to peculiar or unusual restrictions or limiting agreements).

3.9.3 Contracting activity furnished software. Software that the contracting activity provides to the contractor to be used for the contracted software effort and included in the DSSE.

3.9.4 Software that is to be developed. Software to be developed, or in any stage of development, that is needed to fulfill the requirements of the contracted effort.

3.10 Target computer system. Computer equipment, software, or procedures which are physically a part of an operational system.

3.11 Unlimited rights. The rights to use, duplicate, or disclose technical data or computer software in whole or in part, in any manner and for any purpose whatsoever, and to have or permit others to do so.

3.12 Use of "shall", "will", "should", and "may". "Shall" is used to express a provision that is binding; "should" and "may" are used to express nonmandatory provisions; "will" is used to express a declaration of purpose or intent.

(Intentionally Blank)

4. GENERAL REQUIREMENTS

(3)

4.1 Software support environment. The contractor shall define, implement and integrate all software and related items that will be used to develop and support the deliverable software required under the contract. The contractor shall identify all software and related items that are recommended by the contractor for use by the designated life cycle software support activity to support the contractually deliverable software throughout its operational life. The contractor shall also identify the approach proposed to ensure and warrant the existence of the capability to perform software support of the contractually deliverable software by the designated life cycle software support activity. The contractor shall submit the proposed approach to the contracting activity and obtain approval from the contracting activity prior to commencing the contracted software effort. (See 6.0)

4.2 Contracting activity furnished resources. The contracting activity may designate resources to be used by the contractor. The contractor shall identify to the contracting activity any resources expected to be furnished by the contracting activity to support the contracted effort. These resources shall be identified in the contractor's proposed DSSE approach. The contracting activity retains the option to furnish the resources or to require, through the contract, the contractor to obtain them. (See 6.0)

4.3 Rights in documentation and computer software. The contractor may propose the use or delivery of software and documentation with limited or restricted rights, or other potential licensing agreements. Any such contractor proposals must clearly identify for each item the expected economic and other benefits or risks to accrue to the contracting activity and the expected constraints on the rights of the contracting activity. Unless prior approval for the use or delivery of this software is obtained from the contracting activity, the contractor shall ensure that the contracting activity shall have unlimited rights in all computer software, equipment, and documentation that is required to evaluate, generate, install, integrate, test, modify, support, and operate the contractually deliverable software. All such items necessary to ensure the performance of these functions shall be available for delivery by the contractor to the contracting activity. The contractor shall obtain contracting activity approval prior to implementation or use, and prior to any contractor licenses or agreements associated with the purchase or use of, any commercially available or privately developed software and documentation related to the performance of the contract. (See 6.0)

(Intentionally Blank)

4.4 Deviations and waivers. All resources required to satisfy the requirements of this Standard shall be developed and delivered in complete conformance with the requirements of this Standard, unless a deviation or waiver for each specific item has been previously processed and approved by the contracting activity. The extent of any variance from exact conformance to all applicable requirements shall only be that which is specifically authorized by formally approved deviations and waivers.

(Intentionally Blank)

5. DETAILED REQUIREMENTS

5.1 DSSE (Developmental Software Support Environment). The contractor shall implement a DSSE that provides a full range of engineering and other functional services for the development and support of contractually deliverable software. The contracting activity may designate a specific life cycle software support activity or concept and, additionally, may direct the use of existing LCSSE resources. The contractor shall evaluate alternative methods of providing a DSSE which provides the requisite support services and which is completely compatible with the LCSSE that may have been designated by the contracting activity. The contractor's evaluations shall address, as a minimum, the requirements specified in the following paragraphs and shall identify how the required software support capability within the contracting activity designated LCSSE will be ensured and warranted to the contracting activity. The contractor shall design a DSSE that satisfies all specified requirements and that is fully compatible with the LCSSE that may have been designated by the contracting activity. The contractor shall obtain contracting activity approval of the proposed DSSE approach prior to its implementation or use in performing the contracted software effort. (See 6.0)

5.1.1 DSSE approach. The DSSE approach shall be based on developing and supporting all contractually deliverable software in an environment that has extensive support software resident in a host computer system.

5.1.2 DSSE identification. Unless otherwise specified by the contracting activity, the contractor may propose to utilize the resources of the designated life cycle software support activity, to utilize the contractor's internal software development resources, or to use a combination of those resources. The contractor shall ensure that any recommendation to incorporate commercially available or privately developed software considers the potential economic commitments (initial and recurring), the risks of long term dependence on the subcontractor or vendor, the probability of obsolescence, and the projected stability of the proposed software. (4) The contractor shall identify alternatives, with supporting economic analyses, to provide the capabilities of the commercially available or privately developed software through other means, such as redeveloping or modifying other software. The contractor's approach shall clearly identify the interfaces with any LCSSE designated by the contracting activity. The contractor shall reconcile the operations and support requirements identified by the contracting activity with the proposed DSSE. Once it has been approved by the contracting activity, any changes in the contractor's DSSE shall be subject to contracting activity approval prior to implementation or use.

(7)

(Intentionally Blank)

5.1.3 DSSE contents. The DSSE shall provide, as a minimum, a set of defined user/system interfaces, a set of software support tools, and a central library for the storage of software and all information associated with the development and support of the contractually deliverable software over its life cycle. The DSSE shall provide for storage of software both in a source form and in a form that has been compiled for a host computer system or a particular target computer system. The DSSE shall include a control language which presents an interface to the user and to the information in the central library. The software support tools shall include tools for software development, testing, support, maintenance, and configuration control. The DSSE shall support the functions of project management, documentation, and release control. The contracting activity may specify specific data bases, tools, interfaces, and procedures for inclusion in the DSSE.

5.1.4 DSSE operation. The contractor shall establish procedures and controls for access, use, generation, and change of all software in the DSSE. As a minimum, the contractually specified software development requirements for library usage and control, software generation, software operation, software configuration management, software quality assessment, and software trouble reporting shall be included and shall be applied to all software in the DSSE. (5)

5.1.5 Differences between the DSSE and the designated LCSSE. The contractor shall describe all differences between the DSSE and the designated LCSSE. The contractor shall describe all additions to the designated LCSSE, both software and procedures, that are necessary to ensure the compatibility of the DSSE with the designated LCSSE. The contractor shall identify the proposed additions as those that are either required to support a specific application for a particular target computer system or those that have potential for broader usage in the designated LCSSE. (6) The contractor shall also separately identify and justify all software or procedures intended for use in the DSSE, but not recommended for inclusion in the designated LCSSE. For each such item, the contractor shall provide reasons why these additions are not recommended. The contracting activity may specify software or procedures to be added to the designated LCSSE. No contractually deliverable software shall be dependent on any software or procedures that are not deliverable to, or designated by the contracting activity. (7) The additions to the designated LCSSE are subject to approval of the contracting activity prior to implementation or use of the DSSE.

(8)

(Intentionally Blank)

5.1.6 Software source identification. The components of the proposed DSSE may come from four sources, i.e., software that is furnished by the contracting activity, software that is commercially available, software that is privately developed or software that is to be developed under the contract. These software sources are defined in paragraph 3.9 of this Standard. The contractor shall identify the proposed source(s) for all the software to be included in the DSSE. The proposed software sources shall be subject to approval by the contracting activity prior to implementation or use, and prior to any contractor licenses or agreements associated with the purchase or use of, any commercially available or privately developed software.

5.2 DSSE implementation. Upon approval by the contracting activity, the contractor may implement the proposed DSSE. The following paragraphs contain specific requirements for the software in the DSSE that will originate from each of the sources defined in paragraph 3.9.

5.2.1 Software furnished by the contracting activity. The contractor shall manage the software furnished by the contracting activity in accordance with the following paragraphs:

5.2.1.1 Software integration requirements. The contractor shall integrate the contracting activity furnished software with the approved DSSE. Any additions or changes required to the contractor's DSSE to integrate the contracting activity furnished software shall be separately identified, developed, and controlled as required in other parts of this Standard. The contractor's DSSE shall be designed to ensure the independence of the contracting activity furnished software from the rest of the DSSE.

(80) 5.2.1.2 Software documentation requirements. The contractor shall not change the contracting activity furnished specifications or descriptive documentation without prior approval and direction by the contracting activity. The contractor shall fully define and document all additions or changes to the DSSE that were required to properly integrate the contracting activity furnished software. The documentation and delivery requirements for these additions or changes shall be as specified in the contract or in the Contract Data Requirements List.

5.2.1.3 Software quality assessment requirements. The contractor shall include in the contracting activity approved software quality assessment program the procedures necessary to ensure that the requirements for integration of the contracting activity furnished software with the DSSE are satisfied.

(Intentionally Blank)

5.2.1.4 Software configuration management requirements. The contractor shall include in the contracting activity approved software configuration management program the procedures necessary to prevent unauthorized changes to the contracting activity furnished software. The contractor shall identify any problems encountered in the integration and use of this software with the DSSE and shall provide recommended actions to correct these problems to the contracting activity.

5.2.1.5 Software changes. The contractor shall not make any changes to any software furnished by the contracting activity.

5.2.1.6 Software acceptance requirements. The contractor shall ensure that the configuration of the contracting activity furnished software has not been changed and continues to conform with the contracting activity furnished specifications and documentation.

5.2.2 Software that is commercially available. The use of commercially available software shall be subject to contracting activity approval prior to incorporation or use, or prior to any contractor licenses or agreements associated with the purchase or use. The contractor shall identify any licenses or similar agreements by the contractor or among the contractor and subcontractors, vendors, or other sources that will impose any constraints on the use of this software by the designated life cycle software support activity, or by any agent employed by the designated life cycle software support activity to perform life cycle software support of the software developed or delivered under the contract. (9) Unless prior approval to the contrary is obtained from the contracting activity, the contractor shall ensure that the contracting activity shall have unlimited rights to this software. Approval to use this software shall not relieve the contractor of obligations to integrate this software into the DSSE and to ensure compatibility with the designated LCSSE.

5.2.2.1 Software integration requirements. The contractor shall ensure that the commercially available software is properly integrated into the DSSE and will be compatible with the designated LCSSE. Where subcontractor or vendor supplied documentation is used to verify performance, the contractor shall either certify the sufficiency and accuracy of the documentation and test results or accomplish added testing as may be specified by the contracting activity. The contractor shall integrate this software into the DSSE such that any future deficiency corrections or enhancements submitted or released by the original supplier of the software can be readily incorporated by the contracting activity. (10) For commercially available software that is unique to the target computer system, the contractor's DSSE shall be designed to ensure the independence of this software from the rest of the DSSE and the designated LCSSE.

(Intentionally Blank)

5.2.2.2 Software documentation requirements. The documentation and delivery requirements for the commercially available software shall be as specified in the contract or in the Contract Data Requirements List. Where existing documentation satisfies the intent of this Standard, and modification or redevelopment of the documentation is not cost-effective or intended, the existing documentation may be substituted, subject to prior approval by the contracting activity. (See 6.0)

5.2.2.3 Software quality assessment requirements. The contractor shall apply the contracting activity approved software quality assessment program to the commercially available software. The contractor shall include in the contracting activity approved software quality assessment program the procedures necessary to ensure that this software satisfies its specified requirements and is properly integrated into the DSSE.

5.2.2.4 Software configuration management requirements. The contractor shall apply the contracting activity approved software configuration management program to the commercially available software. The contractor shall include in the contracting activity approved software configuration management program the procedures necessary to prevent unauthorized changes to this software. The contractor shall identify any problems encountered in the integration and use of this software with the DSSE and shall provide recommended actions to correct these problems to the contracting activity.

(11) 5.2.2.5 Software changes. The contractor shall not make any changes to the commercially available software without prior approval of the contracting activity. If any of this software must be changed from its commercially available version or release, it shall be recategorized and managed from that point on as software that is to be developed. The contractor shall be responsible for identifying and resolving with the original supplier of the software any deficiencies or incompatibilities of this software with both the DSSE and the designated LCSSE. The contractor shall identify to the contracting activity all changes submitted or released by the original supplier of the software, with an assessment of the possible effects of incorporation in the DSSE and the designated LCSSE. The contracting activity may designate changes submitted or released by the original supplier of the software for incorporation in the DSSE and the contractor shall implement all such designated changes into the DSSE.

5.2.2.6 Software acceptance requirements. In addition to any criteria specified by the contracting activity, commercially available software acceptance shall be predicated upon compatibility with the designated LCSSE and satisfactory resolution of any limited or restricted rights issues.

(Intentionally Blank)

5.2.3 Software that is privately developed. The use of privately developed software, whether supplied by the contractor, subcontractors or vendors, or any other source, shall be subject to contracting activity approval prior to incorporation or use, or prior to any contractor licenses or agreements associated with the purchase or use. Unless prior approval to the contrary is obtained from the contracting activity, the contractor shall ensure that the contracting activity shall have unlimited rights to this software. The contractor shall identify any licenses or similar agreements by the contractor or among the contractor and subcontractors, vendors, or other sources that will impose any constraints on the use of this software by the designated life cycle software support activity, or by any agent employed by the designated life cycle software support activity to perform life cycle software support of the contractually deliverable software. Approval to use this software shall not relieve the contractor of obligations to integrate this software into the DSSE and to ensure its compatibility with the designated LCSSE.

REFER TO (1)

5.2.3.1 Software integration requirements. The contractor shall ensure that the privately developed software is properly integrated into the DSSE and will be compatible with the designated LCSSE. Where existing documentation is used to verify performance, the contractor shall either certify the sufficiency and accuracy of the documentation and test results or accomplish added testing as may be specified by the contracting activity. The contractor shall ensure that any recommendation to incorporate privately developed software considers both the life cycle economic and other benefits or risks to the contractor and the contracting activity. The recommendation should include an assessment of the software's and documentation's quality, the lost or impaired capabilities that would result if the software is not used, and the effort required to develop or modify added software or documentation to provide similar capabilities. For privately developed software that is unique to the target computer system, the contractor's DSSE shall be designed to ensure the independence of this software from the rest of the DSSE.

(12)

5.2.3.2 Software documentation requirements. The documentation and delivery requirements for the privately developed software shall be as specified in the contract or in the Contract Data Requirements List. Where existing documentation satisfies the intent of this Standard, and modification or redevelopment of the documentation is not cost-effective or intended, the existing documentation may be substituted, subject to prior approval by the contracting activity. (See 6.0)

(Intentionally Blank)

5.2.3.3 Software quality assessment requirements. The contractor shall apply the contracting activity approved software quality assessment program to the privately developed software. The contractor shall include in the contracting activity approved software quality assessment program the procedures necessary to ensure that this software satisfies its specified requirements and is properly integrated into the DSSE.

5.2.3.4 Software configuration management requirements. The contractor shall apply the contracting activity approved software configuration management program to the privately developed software. The contractor shall include in the contracting activity approved software configuration management program the procedures necessary to prevent any unauthorized changes to this software. The contractor shall identify any problems encountered in the integration and use of this software with the DSSE and shall provide recommended actions to correct these problems to the contracting activity.

5.2.3.5 Software changes. The contractor shall not make any changes to the privately developed software without prior approval of the contracting activity. All changes proposed by the contractor to this software shall identify the impact of the change on the contractually deliverable software, the DSSE, and the designated LCSSE. The contractor is responsible for identifying and resolving with subcontractors, vendors, or other sources any deficiencies or incompatibilities of this software with both the DSSE and the designated LCSSE. The contractor shall identify to the contracting activity all changes submitted or released by the original supplier of the software, with an assessment of the possible effects of incorporation in the DSSE and the designated LCSSE. The contracting activity may designate changes submitted or released by the original supplier of the software for incorporation in the DSSE and the contractor shall implement all such designated changes into the DSSE.

5.2.3.6 Software acceptance requirements. In addition to any criteria specified by the contracting activity, privately developed software acceptance shall be predicated upon compatibility with the designated LCSSE and satisfactory resolution of any limited or restricted rights issues.

5.2.4 Software that is to be developed. All support software in this category shall be developed in accordance with the contractually specified software development requirements.

(Intentionally Blank)

5.2.4.1 Software integration requirements. The contractor shall ensure that the software to be developed is properly integrated into the approved DSSE and will be compatible with the designated LCSSE. The contractor shall include all necessary testing as part of the overall software and system test program. The contractor shall design the software in this category for compatibility with, and operation in, the designated LCSSE. The contractor shall completely identify all adaptations or changes to this software to accommodate any differences between the DSSE and the designated LCSSE. The design of this software shall isolate and identify all DSSE and designated LCSSE dependencies.

5.2.4.2 Software documentation requirements. The documentation requirements for the software in this category shall be as specified in the contract or in the Contract Data Requirements List.

5.2.4.3 Software quality assessment requirements. The contractor shall apply the contracting activity approved software quality assessment program to the software in this category to ensure that it is developed according to contractual requirements. The software quality assessment program shall be supplemented as necessary to ensure that the requirements herein for compatibility of the software with the designated LCSSE are satisfied.

5.2.4.4 Software configuration management requirements. The contractor shall apply the contracting activity approved software configuration management program to the software to be developed. The contractor shall include in this program any added documentation and configuration management requirements that have been specified by the contracting activity.

5.2.4.5 Software changes. The contractor shall establish internal baselines for this software in accordance with the contractually specified software development requirements. After the internal baselines have been established, all changes to this category of software proposed by the contractor shall additionally identify the impact of the change on the operational software, the DSSE and the designated LCSSE.

5.2.4.6 Software acceptance requirements. In addition to any criteria specified by the contracting activity, acceptance of the developed software shall be predicated upon compatibility with the designated LCSSE.

(Intentionally Blank)

5.3 Establishment of software supportability within the designated life cycle software support activity. In addition to any other requirements specified by the contracting activity, final acceptance of the contracted software effort shall be predicated on establishment of a satisfactory support capability for the contractually deliverable software in the designated life cycle software support activity. The required support capability shall include the compatibility of the DSSE with the contracting activity designated LCSSE, and the capability of the designated LCSSE to perform software support for the contractually deliverable software. The contractor shall define for contracting activity approval the proposed approach for ensuring and warranting the required support capability. The methods used to satisfy these requirements, as a minimum, are specified in the following paragraphs. (See 6.0)

5.3.1 Identification of additions to the designated LCSSE. The contractor shall identify all software and procedures in the DSSE that are required by the contracting activity to properly support the contractually deliverable software throughout its life cycle. The contractor shall describe how any additions of software and procedures from the DSSE will interface with the existing software and procedures in the designated LCSSE. (See 6.0)

5.3.2 Description of the designated LCSSE operation. The contractor shall describe how the designated LCSSE shall be used to evaluate, generate, install, integrate, test, modify, and operate the contractually deliverable software. The contractor shall describe the procedures required by the designated LCSSE to accomplish performance evaluation, software and system generation, development and testing of changes, simulation, emulation, training, software integration, configuration management, and distribution for the contractually deliverable software. (See 6.0)

5.3.3 Implementation of additions to the designated LCSSE. The contractor shall plan for and implement the transfer of software support for the contractually deliverable software to the designated life cycle software support activity. This effort shall be designed to ensure a phased transfer without loss or degradation of the support required for the delivered software or to other tasks currently performed by the designated life cycle software support activity. The contractor shall identify the requirement for use of any contracting activity resources during the transfer phase. The contractor shall plan lead-time to ensure completion of the transfer prior to activation of the first operational target computer system

(Intentionally Blank)

or prior to the planned assumption of software support responsibilities by the designated life cycle software support activity. The contractor shall ensure that the procedures for operation of the designated LCSSE completely describe all methods necessary to evaluate, generate, install, integrate, test, modify, and operate the contractually deliverable software. The contractor shall make available assistance to support the resolution of any problems encountered under operation by the designated life cycle software support activity personnel during the transfer period and during a period of time subsequent to the transfer as specified by the contracting activity. These procedures shall be subject to approval of the contracting activity prior to implementation. (See 6.0)

5.3.4 Supportability and compatibility requirements. The contractor shall implement the contracting activity approved approach to ensure and warrant that the DSSE is completely compatible with the designated LCSSE, and shall ensure that the designated LCSSE has the capability to perform software support for the contractually deliverable software. The procedures shall be subject to approval of the contracting activity prior to implementation. (See 6.0) The satisfaction of the supportability and compatibility requirements shall depend on the existence of the following conditions:

a. All contractually deliverable software is capable of being evaluated, generated, installed, integrated, tested, and modified utilizing only the contracting activity designated and contractor delivered software in the designated LCSSE.

b. All operations or functions accomplished by the contractor's DSSE, and identified to or by the contracting activity for inclusion in the designated LCSSE, can be performed in the designated LCSSE.

c. The delivered software will produce identical results when operated in the target computer system, whether generated in the contractor's DSSE or generated in the designated LCSSE.

(Intentionally Blank)

6. MISCELLANEOUS

Contract data requirements. When this standard is used in an acquisition which incorporates the DD Form 1423, Contract Data Requirements List (CDRL), the data requirements identified below shall be developed as specified by an approved Data Item Description (DD Form 1664) and delivered in accordance with the approved CDRL incorporated into the contract. When the provisions of DOD FAR Supplement, Part 27.410-6, are invoked, and the DD Form 1423 is not used, the data specified below shall be delivered by the contractor in accordance with the contract or purchase order requirements. Deliverable data required by this standard is cited in the following paragraphs:

Paragraph no.	Data requirements title	Applicable DID no.	Options
4.1, 5.1, 5.3, 5.3.1, 5.3.3, 5.3.4	Developmental Software Support Environment Plan	DI-E-7140	None
5.2.2.2, 5.2.3.2	Documentation of Commercially Available or Privately Developed Software	DI-E-7141	None
5.3.1, 5.3.3, 5.3.4	Software Support Transition Plan	DI-E-7142	None
5.3.2	Life Cycle Software Support Environment Users Guide	DI-E-7143	None

Review activity:
Army - AV, CR

Preparing activity:
Army - AR
(ECRS-A007)

(Intentionally Blank)

PANEL IV LIST OF BRIEFINGS.

1. DOD-STD-1467 (AR) Military Standard, "Software Support Environment" by Chuck Gordon, CACI, Inc. (703) 276-2838.

(Intentionally Blank)

PDSS MANAGEMENT INDICATORS AND QUALITY METRICS
PANEL V
PROCEEDINGS

OBJECTIVE.

The objective of Panel V, Management Indicators and Quality Metrics, is to identify management indicators and quality metrics applicable to PDSS.

BACKGROUND.

How many software projects can you name which were developed on time, within cost and with a high degree of reliability, usability and maintainability? The answer to that question is the reason for studying the use of potential management indicators and quality metrics to improve PDSS. We are still in the age of software development as an art, not an engineering science. We seem to repeat the same mistakes on project after project, with no end in sight.

But there is an end in sight, for we are running out of money while the number of projects continues to increase. And the excessive costs, schedule slips, poor quality and lack of maintainability cause increasingly more difficult problems in PDSS. By moving the discipline of software engineering from an art to a science, we hope to better control costs, meet our schedules and improve our product quality, thereby directly benefitting the PDSS activity.

It has been evident for a number of years that a more disciplined approach to the management of software, both during the development and PDSS phases, is required. The evidence has been the poor quality of delivered products, the cost and schedule overruns during development and the excessive cost and maintainability problems during PDSS.

What is the difference between an art and an engineering science?

Measurement. In the nineteenth century the famous British physicist, Lord Kelvin, said that:

> When you can measure what you are speaking about, you know something about it. When you are unable to use a quantitative description, then your knowledge is meager and unsatisfactory.

The charter of our panel recognizes that there are two distinct types of measurement required to assist PMs in software development and PDSS: management indicators and quality metrics.

Management indicators are measures of the process of system and software development. They are intended principally for use by management to compare expected versus actual results in such areas as cost and schedule. By comparing expected versus actual measures at major milestones in the development cycle, increasingly accurate assessments can be made concerning the ultimate cost and duration of the development process. This process will allow better informed management decision-making to occur. Without these management indicators, the non-technical manager is often times faced with complex and urgent decisions, with only a bewildering array of detailed technical data to assist him. With management indicators, better decisions will result in better software systems.

In addition, by building an historical data base of such expected versus actual results, we should become much more adept at estimating the true cost and schedule of a future development. In today's competitive and resource-limited development environment, the DOD attempts to minimize cost and schedule estimates found in its RFPs, in order to help justify the necessary congressional budget allocations and "get the most bang for the buck". In turn, contractors attempt to conform to these RFP estimates if they hope to win the awards, even if they do not believe the job can be accomplished properly with the resource levels stipulated in the RFP. Thus, the contractor who wins the award may not be the best contractor for the job, even though they may have been the "lowest bidder". This results in the typical cost and schedule overruns we all see. By providing the DOD with historical data to support more realistic RFP resource estimates, there should be less chance for a mismatch between proposed program cost and actual cost, clearly benefitting the DOD, industry and the U.S. taxpayer.

Quality metrics are measures of the product of system and software development. They are intended principally for use by the technical staff to compare expected versus actual product quality in such areas as number and type of errors, requirements traceability, completeness, and reliability. By comparing expected versus actual quality metrics at major milestones in the development cycle, increasingly accurate assessments can be made concerning the ultimate quality of the development product. Indeed, a quantitative definition of quality will then be available for virtually the first time in software development for DOD systems. Quality metrics will allow better informed technical (and management) decision making to occur. The technical manager is faced with decisions such as, "Is our design complete?", "Have we tested enough?", "Have we met our quality goals?", and "Will the system be reliable and available under stress?". With the proper application of quality metrics, better decisions will result, which in turn will produce more maintainable systems of better quality and lower cost.

Possibly the most beneficial aspect of the use of indicators and metrics is its "carryover" benefit from development to PDSS. Indicators and metrics aid PDSS by allowing assessment of the initial PDSS system component baseline status. Subsequently, changes to this baseline can be assessed for possible impacts on quality, maintainability and testability. This will allow estimation and allocation of limited PDSS resources to aid future maintainability and enhancement.

In addition, by building a multiservice* central repository of such expected versus actual results, we should become much more adept at achieving the desired (i.e., quantifiable) quality of future DOD software systems. The actual form of implementation of the central repository (e.g., per Service; per PDSS agency; etc.) is left to JLC discretion. In today's complex and challenging weapons systems environment, the DOD has virtually no means of quantitatively specifying and measuring the quality of its multimillion dollar systems. In turn, contractors and the DOD acquisition program managers are under cost and schedule pressure to produce systems, and are not primarily driven by the quality requirements of a system specification. In many instances, the contractor and/or the DOD program manager makes decisions based on shortening schedules and cutting costs, rather than on improving the system's quality. This results in the inconsistent quality we see in our weapons systems. By providing the DOD with historical quality measurement data, the added visibility of measurable quality will only serve to improve the attained level of quality in DOD systems, to the great benefit of the PDSS activities.

CHALLENGE.

The central theme of Orlando II is "Solving the PDSS Challenge." The workshop addressed various aspects of PDSS to identify areas which offer significant payoffs in terms of cost reduction, improved system reliability, streamlining of the PDSS process, and, most importantly to Panel V, incorporating practical management indicators and quality metrics into the governing software development standards (e.g., DOD-STD-2167). The challenge for Panel V, as discussed in the Orlando II Master Plan, is described as "... to identify management indicators and quality metrics applicable to PDSS." Specifically, our objectives are to:

1. Identify a standard set of management indicators that support a management assessment of the software development process.

---

* The term multiservice refers to Joint Army/Navy/Air Force/Marine Corps.

2. Identify a standard set of quality metrics that support technical assessment of software product quality.

3. Recommend the best approach for incorporating the above measures into the DOD software development/PDSS process, so that all future PDSS software systems benefit from their use.

ASSUMPTIONS.

In forming the recommendations of Panel V, the following basic assumptions have been made:

1. A reasonable, tailorable set of management indicators to define "product status" in quantitative terms is achievable (e.g., cost, schedule, life cycle factors, etc., such as AFSCP 800-43).

2. A reasonable, tailorable set of measures to define "product quality" in quantitative terms is achievable (e.g., number and types of errors, requirements traceability, testability, etc.).

3. Viable ways to incorporate the use of these indicators and metrics into the DOD development cycle can be found, without undue cost or schedule impacts.

4. A method for centralizing the gathering of multiservice historical indicator and metric data (and associated tools) for use on future DOD projects should be pursued. Thus, we will be building our future knowledge on a solid foundation of quantified past experience. Again, the JLC should decide how to best implement this recommendation (e.g., per Service).

5. Allow for the incorporation of modifications, additions and deletions to our recommended minimum sets of measures to continue to refine our measurement process and products.

APPROACH.

The following sections contain detailed recommendations for implementing an indicator/metric program. Some general lead-in issues are discussed first:

1. There is striking similarity in the top level findings of all the Orlando II panels, on such issues as:

   a. Multiservice oversight and leadership.

   b. Need for standardization and policy revision.

   c. Need for the ability to tailor implementation as required.

d. Need for a multilevel training program.

e. Need for top-to-bottom DOD commitment.

2. This set of recommendations requires an outlay of "up front" money. The expense will be recovered and the benefits will far outweigh the cost. Just as the up front cost of insurance for cars, homes and hospital bills is warranted, so is the cost of "metrics" insurance. How much longer can we afford not to have this added protection?

3. Software system engineering as a discipline is in its infancy. Compare the thousands of years of existence of medicine or architecture with our forty-odd years of software systems. Maturity will take time and iterative refinement. So it is with software metrics and indicators. We do not claim to have "metrics perfection and precision". In fact, the current state of metrics permits decision support only, not decision control. As we gradually refine our methods, more exact meaningful measures will result. But we can start to reap the benefits today.

4. Tom Clancy, the author of the recent bestselling Cold War thriller, Hunt For Red October, said in an interview that in the next war, the side with an extra 5% of battle information will win, because this represents a decisive edge in combat. Along the same lines, the purpose of our metrics and indicators is to give PMs, developers and PDSS managers "that extra 5%" of information in the battle to produce quality software systems on time and within budget. This may just be the decisive edge we need to win our PDSS battle as well. Panel V's recommendations follow in priority order. We have included a brief title, a paragraph description and such items as cost, schedule, ROI, difficulty and benefits for each recommendation made.

RECOMMENDATIONS.

Primary Recommendation of Panel V. The JLC JPCG-CRM establish a Multiservice Multiphase Management Indicators and Quality Metrics Advocacy Program. Such a program should address the entire spectrum of policies, standards, guidelines, issues and activities that are necessary in applying such management indicators and quality metrics to DOD system developments. Without such high level direction, the following individual recommendations will lack the urgency necessary to insure they are implemented across each component of the DOD. Without the adoption and implementation of the recommendations that follow, the "black art" of software development we now typically pursue will not become a true engineering discipline in any of our lifetimes. Our grandchildren and their children will continue to pay for the same type of redundancy, waste and poor quality that we all must put up with today.

The multiphase program recommended by Panel V is described in brief in the three one-year phases that follow. The individual numbered recommendations shown in each phase are then described in detail after that.

MANAGEMENT INDICATORS AND QUALITY METRICS PROGRAM
June 87 -- PHASE I -- May 88

REQUIREMENTS DEFINITION. The JLC JPCG-CRM should:

1. (KEY) Establish a JLC JPCG-CRM Management Indicators and Quality Metrics Subgroup.

2. (KEY) Mandate the use of the following existing guidelines as a foundation for tailorable indicator/metric usage and future refinement:

   - AFSCP 800-14 (Quality)

   - AFSCP 800-43 (Management)

3. (KEY) Revise existing MCCR policy, guidance and standards and associated DIDs to recommend the use of approved management indicators and quality metrics (see #2 above).

4. Require the option to use (and share across programs and Services) existing government owned automated tools to generate and use such metrics data (e.g., AMS, MSAT, CAT, FASP, SPAR, AMAT, etc.).

5. (KEY) Add error severity code levels to the Software Problem Report (e.g., per the MIL-STD-1679A five error types).

6. Provide a mechanism for users of the above (i. e., policy, standards, guidelines and tools) to provide feedback to the JLC JPCG-CRM subgroup for the purpose of revising, refining and improving the use of management indicators and quality metrics.

BENEFITS.

- An ongoing program (which is not dependent on an individual or group of individuals) which will refine and improve itself over time.

- Immediate results/indicators to project management.

- Higher level of visibility into the status of software development efforts and the quality of software products.

(Intentionally Blank)

DEVELOPMENT. The JLC JPCG-CRM should:

7. Improve the software quality factor definitions currently found in DOD-STD-2167 and associated DIDs.

8. (KEY) Establish a PDSS to Development Agency feedback loop to report any observations on the use of newly developed systems in the field, for the purpose of improving and refining methods for the future.

9. (KEY) Develop a multilevel management indicators and quality metrics guidebook, for the purpose of promulgating the use of these tools in the DOD. The multilevels should include project management, technical management and technical support at a minimum.

10. (KEY) Establish an R&D activity via Rome Air Development Center (RADC), SEI or similar appropriate agency which addresses management indicators and quality metrics issues.

11. (KEY) Re-emphasize the importance of the requirements definition phase of the software system development/ maintenance/revision life cycle by mandating traceability of requirements to documentation, code and test cases. Automation of this traceability process is desirable where size and complexity warrant this approach. This will help insure completeness, testability and usability of the new system. Emphasis on appropriate quality factors as requirements is suggested. In addition, strongly consider the use of rapid prototyping in cases where the system requirements document is not yet approved and baselined, to mature the system requirements before proceeding to the design phase.

BENEFITS.

- Cost savings.
- Management visibility.
- Higher quality software.

(Intentionally Blank)

MANAGEMENT INDICATORS AND QUALITY METRICS PROGRAM
June 89 -- PHASE III -- May 90

FULL IMPLEMENTATION. The JLC JPCG-CRM should:

12. (KEY) Mandate the use of approved management indicators and quality metrics in development RFPs. Provide standard contract clauses and model SOWs which invoke standards and guidelines for metrics, indicators, related automated tools, etc.

13. (KEY) Establish a multilevel, multiservice training program for management indicators and quality metrics to perpetuate the proper use of same in the DOD.

14. (KEY) Establish a multiservice central data/tools/references bank for management indicators and quality metrics. The data tools and references in this bank would be accessible to all in the DOD for use in gathering data, estimating, identifying trends and furthering research in the field.

BENEFITS.

- External acceptance.
- Schedule and budget improvements.
- Joint Service visibility.

(Intentionally Blank)

INDIVIDUAL RECOMMENDATIONS.

The recommendations of Panel V are given below and listed in order of priority for implementation.

1. (KEY RECOMMENDATION) (Recommendation 4-5-01). THE JLC MUST TAKE THE INITIATIVE TO ESTABLISH AND FUND A JLC JPCG-CRM AND PROGRAM TO FOSTER THE USE OF APPROVED MANAGEMENT INDICATORS AND QUALITY METRICS.

Description. Currently, management indicators and quality metrics are not being used consistently, even within individual programs within individual Services, to improve cost, schedule and product quality. This is true even though there are a number of examples of successful use of management indicators and quality metrics on specific programs within each of the Services.

This panel proposes that JLC JPCG-CRM establish a full time joint Service CRM Subgroup with the responsibility of identifying, implementing, maintaining and fostering the use of management indicators and quality metrics throughout DOD. Pilot projects could be identified and data collection could begin for newly identified projects 60-90 days after the subgroup has been established. In parallel with data collection, such activities as data feedback, data analysis, training, the generation of draft DOD management indicators and a draft DOD policy should follow. As an end result, a comprehensive DOD policy, new or improved standards and a final set of management indicators can be developed and published.

Methodology should be established to include data collection methods, analysis philosophy, automated tool support and a maintenance policy for data and tools. Inherent in this methodology will be the incorporation of feedback into policy, standards, and the metrics and indicators themselves.

BENEFITS.

There is a software crisis with regard to cost, schedule and product quality. Management indicators and quality metrics provide managers and engineers with visibility into the software product and process. This visibility aids in the detection and isolation of problems in the software, in the monitoring of consistent and disciplined use of accepted management and engineering techniques, and enhances the management of cost, schedule, and quality. A central approach is required to transition the existing metric research into widespread practice within the joint Services. There is a need to foster the identification, application, quantification, calibration, implementation, and standardization of management indicators and quality metrics.

2. (KEY RECOMMENDATION) (Recommendation 4-5-02). MANDATE THE USE OF APPROVED TAILORABLE MANAGEMENT INDICATORS AND QUALITY METRICS.

Description. Orlando II Panel V recommends the following existing guidelines to encompass the tailorable indicator/metric set to build upon today:

- "AFSC Software Management Indicators - Management Insight", AFSCP 800-43, 31 Jan 86

- "AFSC Software Quality Indicators - Management Quality Insight", AFSCP 800-14, 24 Sep 86

The following two technical reports (TRs) were recommended for containing additional information on useful metrics, but there was no consensus on these documents being part of the initial standard set of management indicators and quality metrics:

- "Specification of Software Quality Attributes", RADC TR 85-37, 1985

- "Methodology for Software Reliability Prediction and Estimation", RADC TR XX-XX, 1987.

There was unanimity on Panel V in this recommendation. Even though there is undoubtedly room for improvement in the above guidelines, they provide a firm foundation for current application and future refinement. Future procurements could cite the above and tailor the indicators/metrics to suit individual projects. The feedback solicited/received from such project developments would permit refinement and enhance utility.

3. (KEY RECOMMENDATION) (Recommendation 4-5-03). REVISE APPROPRIATE DOD STANDARDS AND DIDs TO INCORPORATE APPROVED MANAGEMENT INDICATORS AND QUALITY METRICS.

Description. The following existing DOD standards and guidelines (at a minimum) should be reviewed and revised to incorporate/ mandate/require/tailor the use of JLC JPCG-CRM approved management indicators and quality metrics:

- o DOD-STD-2167 (Software Development of Mission Critical Computer Systems)

- o DOD-STD-1521B (Technical Reviews and Audits)

- o MIL-STD-490A (Specification Practices)

- o MIL-STD-480A, 481A, 482A, and 483A (Configuration Management)

- o MIL-STD-499 (System Engineering Management)
- o MIL-STD-1679A (precursor to DOD-STD-2167 and still used for new/enhanced versions of existing systems in PDSS)
- o MIL-STD-1369 (Integrated Logistics Support)
- o MIL-S-52779A (Software Quality Assurance Program Requirements--to be superseded by DOD-STD-2168 (DRAFT), but still used for new/enhanced versions of existing systems in PDSS)
- o DOD-7935.1S (Automated Data Systems Documentation Standards)
- o Etc.

4. (Recommendation 4-5-04). REQUIRE THE USE OF EXISTING GOVERNMENT-OWNED AUTOMATED INDICATOR/METRIC TOOLS.

Description. There are in existence today software tools that store, analyze, and partially automate the collection of the data that comprise management indicators and quality metrics. The use of these tools simplifies the costs associated with collecting and analyzing management indicators and quality metrics. Among the tools that are available are: The Automated Measurement System (RADC/COEE), Multistatic Analyzer Tool (TECOM, Fort Huachuca), Complexity Analysis Tool (AMCCOM), Facility for Automated Software Production - FASP (NADC), Source Program Analyzer and Reporter - SPAR (NRL), and Ada Measurement and Analysis Tool (DRC).

IMPLEMENTATION.

Near Term. These tools should be made available to any DOD project that will be applying software management indicators and quality metrics. Project offices using a tool should evaluate the tool and recommend improvements.

Mid Term. Based upon feedback from near term use, current tools should be improved into production quality tools. Future versions of these tools should also accommodate the collection and reporting of the entire set of JLC JPCG-CRM approved measures.

- The JLC should insure that software engineering development environments produced/acquired by DOD shall address the collection and reporting of management indicators.

BENEFITS.

Automating software measurements (whether by stand alone tools or within software engineering development environments) will greatly ease the transition of the measurements into widespread use. In addition, information will be made available to the program offices in a more timely manner.

5. (KEY RECOMMENDATION) (Recommendation 4-5-05). REVISE DOD-STD-2167, DOD-STD-2168 (DRAFT) AND RELATED DIDs TO INCORPORATE ERROR SEVERITY LEVELS INTO THE ASSOCIATED SOFTWARE PROBLEM REPORT FORMAT.

Description. Revise the appropriate development standards and DIDs to incorporate error severity levels. Panel V recommends a scheme similar to that found in DOD-STD-1679A:

Level	Description
1	Fatal system error
2	One entire system function inoperative
3	One entire system subfunction inoperative
4	Minor code or documentation error
5	Miscellaneous (e.g., misspelling)

This differentiation of error degree would allow more accurate assessment of development status (e.g., 100 existing errors of severity 5 implies a much better development status than the same number of severity 1).

The capturing of this data in a project deliverable document (e.g., the SDP),and [ultimately] the centralization of such program information through the proposed JLC JPCG-CRM Subgroup would support improved PDSS planning and execution. Sharing of quality and estimation data would be promoted:

- Between Development and PDSS agencies.
- To estimate future development/PDSS levels of effort.
- To identify key trends encompassing entire classes of systems (e.g., Command and Control systems).
- To improve the engineering methods of developers and PDSS.
- To conduct research into improving system quality.

We recommend that the following data be captured during program development for inclusion in the Software Development Plan (defined by DID DI-MCCR-80030) and for use during PDSS:

- Total number of errors encountered (a one-page table will do):

  > by severity level (e.g., number of "severity 1" errors - number of errors which would have resulted in system "crashes")

  >> by software life cycle phase (e.g., to assess the effectiveness of error eradication during the "code and unit test" phase, for example)

- The same data above "normalized" to thousands of lines of code (KLOC): e.g., 172 errors in a system of 2,113 lines of code (data lines included) equals 81.4 errors/KLOC found during the entire development cycle (or 172 errors/2.113 KLOC).

6. (Recommendation 4-5-06). THE JLC-CRM SUBGROUP MUST SEEK FEEDBACK FROM USERS OF APPROVED MANAGEMENT INDICATORS AND QUALITY METRICS TO ALLOW REFINEMENT AND SUBSTANTIATION OF SAME.

Description. To speed the process of improvement and enrichment of the original tailorable set of approved management indicators and quality metrics, the JLC JPCG-CRM subgroup needs feedback, both positive and negative, in order to improve the utility of these tools. Without this feedback loop, the management indicators and quality metrics may never achieve a status of usefulness and practicality, which is vital to their universal adoption.

7. (Recommendation 4-5-07). SOFTWARE QUALITY FACTOR TERMINOLOGY DEFINITIONS MUST BE IMPROVED IN DOD-STD-2167 AND ASSOCIATED DIDs.

Description. There is a need to make the terminology for Software Quality Factors more meaningful. This will allow upper management and less experienced technical support to better understand what is being measured during development and PDSS.

Two terms in particular, "Flexibility and Maintainability" should be combined into a new term that could be called Supportability, and defined as:

> "The effort required to enhance or modify software to meet a new requirement, correct latent defects or adapt to a hardware change."

The existing definitions are both contained in the new term. Also, a minority of Panel V felt the term "Maintainability" is misleading because software is not maintained: - software undergoes design changes or defect correction. There is no need with software to do anything just because a certain amount of time has passed (e.g. preventive maintenance), as there is with hardware which degrades over time.

The term "Reliability" should also be redefined for software, or another term should be used. This is because software does not "break or wear out over time" in the same way that hardware does. A software system fails to fully meet changing user needs over time. Software always works as programmed. Latent defects may be discovered over time; however, the software will still function properly for all environments tested, and is available to fulfill mission requirements while the software becomes more mature, not less.

Agreement. This is not a majority opinion; however, it should be brought to the attention of the JLC. We need to avoid the general use of hardware terms for software. These hardware terms may be misleading and inappropriate. That is not to say that hardware concepts cannot be carried over to software with careful consideration and adjustment: the parallel is there, but it is not always exact or completely accurate.

Implementation. Change the next revision of DOD-STD-2167.

Impact of not Implementing. Continued misunderstanding of Software Quality.

BENEFITS.

Everyone in DOD will know what is meant when a given quality factor is used. Such is not the case today, resulting in confusion, misuse and lack of uniformity. This one change will greatly assist the collection of meaningful management indicators and quality metrics data.

8. (KEY RECOMMENDATION) (Recommendation 4-5-08). ESTABLISH A FORMAL FEEDBACK LOOP FROM DOD MAINTENANCE ACTIVITIES BACK TO THE DEVELOPING AGENCIES.

Description. At a minimum, the feedback should describe:

- Which projects were successful (how, why, etc.).
- Which projects were failures (how, why, etc.).
- Lessons learned during maintenance and use.
- Which indicators and metrics were used and results of use.

This concept will help refine development methods, improve future system quality and thereby reduce the strain on PDSS resources. There may be some instances of this feedback already occurring, but it is sporadic and rare.

The main obstacle to this recommendation is the possible perception by development agencies that their work will be unduly criticized. This must be overcome, because the current practices do not foster a long term outlook: developers rarely worry about what happens to the systems after they achieve approval for operational use, simply because there is no incentive for them to do so. This flaw must be corrected.

9. {KEY RECOMMENDATION} (Recommendation 4-5-09). DEVELOP A MULTILEVEL GUIDEBOOK DETAILING THE RECOMMENDED USE OF MANAGEMENT INDICATORS AND QUALITY METRICS.

Description. The JLC JPCG-CRM subgroup should sponsor an effort to develop a management indicators and quality metrics handbook (guidebook) that integrates software management indicators (AFSC Pamphlet 800-43), software quality indicators (AFSC Pamphlet 800-14), software reliability measures (RADC-TR-87-XX, Guidebook for Software Reliability Prediction and Estimates), and software quality measures (RADC-TR-85-37, Specification of Software Quality Attributes).

Guidance for selecting software quality factors important to the success of a project will be addressed as well as guidance for implementing measures throughout the complete life cycle. The guidebook will address such issues as theory, implementation procedures, available automated tools, evaluation techniques, references, points of contact, how to contractually apply the indicators/metrics, how to analyze the collected data, how to interpret the analysis results, etc.

BENEFITS.

- Will provide an information source for implementation, in lieu of a full scale training program.
- Act as a training tool for management indicators and quality metrics.
- Provide managers with information for measurement indicators and quality metrics implementation.
- Provide technical support for measurement indicators and quality metrics implementation.
- Allow uniform implementation of measurement indicators and quality metrics (for example, guiding their use in future RFPs).

- Act as a central source for measurement indicators and quality metrics information.

10. (KEY RECOMMENDATION) (Recommendation 4-5-10) ESTABLISH A COMPREHENSIVE RESEARCH AND DEVELOPMENT (R&D) PROGRAM FOR MANAGEMENT INDICATORS AND QUALITY METRICS.

Description. Such an R&D program should encompass the following aspects, at a minimum:

- o Survey current use of management indicators and quality metrics across each DOD agency to define a base of experience, available automated tools and lessons learned to date.

- o Gather feedback on the use of measurement indicators and quality metrics: successes, failures, suggestions, refinements, new management indicators and quality metrics, etc. This feedback will prove invaluable for refinement, calibration and certification of management indicators and quality metrics, to increase awareness and acceptance across DOD.

- o Procure automated indicator/metric tools and associated documentation; collect government owned tools to allow sharing and automation of the use of management indicators and quality metrics.

- o Study future areas for measurement indicators and quality metrics use (e.g., parallel processors and associated languages, expert systems, robotics, etc.).

- o Support the use of (automated) management indicators and quality metrics in future programming support environments (e.g., Ada's) and support tools.

- o Experiment with new/revised measurement indicators and quality metrics on selected projects, with results propagated DOD wide.

- o Experiment with adding measurement indicators and quality metrics automated functions into preliminary compiler and PDL tools. Commonly used complexity metrics (e.g., Halstead's and McCabe's are public domain) could supply valuable insight into potential problem areas in both design and code at the earliest possible time, thereby reducing cost and maximizing test effectiveness.

11. (KEY RECOMMENDATION) (Recommendation 4-5-11). EMPHASIZE THE REQUIREMENTS DEFINITION PHASE THROUGHOUT THE SOFTWARE SYSTEM LIFE CYCLE.

Description. A much stronger emphasis should be placed on Requirements Definition and Tracking in the System/Software Life Cycle. The major benefits to be derived and potential problems to be avoided dictate this approach. Two implementation methods, both of which are already automated, can be employed:

- o Requirements Traceability. An automated method needs to be employed which traces each top-level requirement down through the successive levels of documentation to user and operator manuals, to source code and to system test cases. This is a cost-effective, accurate way of insuring that the resulting system will be complete, testable, usable and of good quality during operational use. It will also allow carryover of requirements traceability from development to PDSS, thereby:
  - Retaining the original information on how and why the system was built.
  - Allowing "what if" analysis of proposed changes during PDSS.
  - reducing PDSS costs.
- o Rapid Prototyping. An automated method needs to be employed which simulates the operator interaction with a proposed system after the top-level requirements are specified, but before design is begun. This allows more usable systems to be specified with less iteration of design and revision of requirements during development. In turn, this should result in more usable systems requiring reduced PDSS levels-of-effort.

Such automated aids already exist and have proven themselves in actual use. A Panel V member, Mr. Clell Gladson of the Naval Ocean Systems Center in San Diego, California (619/225-7615 or Autovon 933-7615), has experience with such tools and can provide additional information upon request.

Leadership. Need a DOD-wide Requirements Tracing Tool, available to all DOD agencies for use.

IMPLEMENTATION.

Near Term. Build or buy an automated Requirements Tracing Tool or set of specialized tools for requirements tracing.

Mid Term. Develop a DOD standard tool with documentation.

Long Term. Supply a certified Requirements Tracing Tool to any Development/QA/V&V/PDSS activity which asks for it; including documentation, training and support, from a central DOD facility.

BENEFITS.

- o Return on Investment. Pays for itself multiple times during development, test, and operational system use by reducing test time, redesign iterations, allowing "what if" analysis of proposed changes, etc.

- o Life Cycle Costs. Greatly aids PDSS by assuring full implementation of system requirements in code, tests and documentation. Acts as the "bridge" between development and PDSS, allowing standardized way for PDSS to acquire new systems.

12. {KEY RECOMMENDATION} (Recommendation 4-5-12). MANDATE THE USE OF MANAGEMENT INDICATORS AND QUALITY METRICS DURING ACQUISITION AND PDSS PROCUREMENTS OF DOD SOFTWARE.

Description. All agencies/organizations that have software (Computer Software Configuration Item (CSCI)) development, testing and/or support requirements must be able to mandate tailorable software quality factors, criteria and metrics that are reflected in both development and PDSS contracts. Each organization has unique life cycle responsibilities that cause software to be viewed from different perspectives. Quality attributes that are of primary importance to the agency supporting software during the operational phase will probably have different priority rankings to the development organization. If the life cycle support organization cannot influence the RFP and resulting contract, there is the possibility that the CSCI's may not have the right quality attributes "built in" to the software to insure maintainability (modifiability and supportability) throughout the system's software life cycle.

When the correct software evaluation requirements are included in the RFP, there is a subsequent need to evaluate the contractor's proposed compliance with these requirements as a subset to his overall software quality assurance program. Traditionally, government insight into the contractor's complete software quality assurance program is only obtained when the Software Quality Evaluation Plan (SQEP) is submitted some time after contract award.

IMPLEMENTATION.

- o The JTC must stipulate in appropriate documentation, firm requirements for the user, tester, and life cycle support agency to list and prioritize required software

quality attributes and measurements that are desired to be included in the development contract.

- o Quality requirements in RFPs must explicitly specify special "program unique" quality measurements that must be incorporated into the contractor's total software quality assurance program.

- o RFP instructions to contractors must require that a "Draft Software Quality Evaluation Plan " be submitted with the contractor's proposal. This draft SQEP should be incorporated into the Government's evaluation criteria for contract award.

Impact if not Implemented. Continuation of status quo (including the typical horror stories of escalating software life cycle support costs on many large, complex systems).

BENEFITS.

- o Return on Investment. A function of the length of the total system life cycle and expected frequency of software changes.

- o Life Cycle Cost Impacts. Substantial savings in total life cycle cost if software is designed to contain required attributes that facilitate support during the operational phase of the life cycle.

  The development of software that enhances reliability, feasibility, usability and maintainability attributes will result in both a higher probability of mission success and lower total life cycle software support costs.

Incorporating the contractor's "Draft SQEP" as part of the contract award process provides early insights into the total scope of the contractor's quality program. At little additional cost, the following benefits are achievable:

- Emphasizes the importance the Government is placing on the development of quality software

- Forces the contractor to commit to a fully compliant software quality assurance program early. Undoubtedly, the contractor will propose the strongest possible measurement indicators and quality metrics program to enhance his competitive position.

13. {KEY RECOMMENDATION} (Recommendation 4-5-13). ESTABLISH A MULTI-LEVEL MEASUREMENT INDICATORS AND QUALITY METRICS TRAINING PROGRAM.

Description. Current management indicators and quality metrics technology exists, but is being practiced inconsistently within the DOD. A key aspect of this is the awareness of this technology by acquisition managers and PDSS management.

IMPLEMENTATION.

Near Term. The JLC should implement an education program of measurement indicators and quality metrics concepts. This education program would consist of briefings to high level managers introducing them to the concepts and benefits of management indicators and quality metrics. Managers to be briefed would include acquisition managers, PDSS managers, procurement managers, and upper level technical management. These briefings would be augmented with more technical courses on theory, implementation, tools and evaluation methods for technical support.

Mid Term. The use and benefits of software quality factors should be incorporated into service schools that teach computer resources development and acquisition. Incorporation in initiatives such as Project "BOLD STROKE" in the Air Force, and throughout DOD thereafter, is also highly recommended.

BENEFITS.

- The education program, at a minimum cost, will provide insight into the benefits of using management indicators and quality metrics as effective tools to help manage software.
- Reduced life cycle cost will result through more informed management.

14. {KEY RECOMMENDATION} (Recommendation 4-5-14). ESTABLISH A MULTISERVICE MEASUREMENT INDICATORS AND QUALITY METRICS CENTRAL DATA/TOOLS/REFERENCES BANK.

Description. The JLC should mandate the collection of quality metric data, and deliver this data to the central management indicators and quality metrics data bank. This process of collection should use automated tools to the maximum extent possible. All tools shall also be deliverable. The metric data may be used by acquisition agencies to assure higher quality products. The PDSS agency will use the data to refine their resources requirements and to track software activity through life cycle completion. See Figure 5.

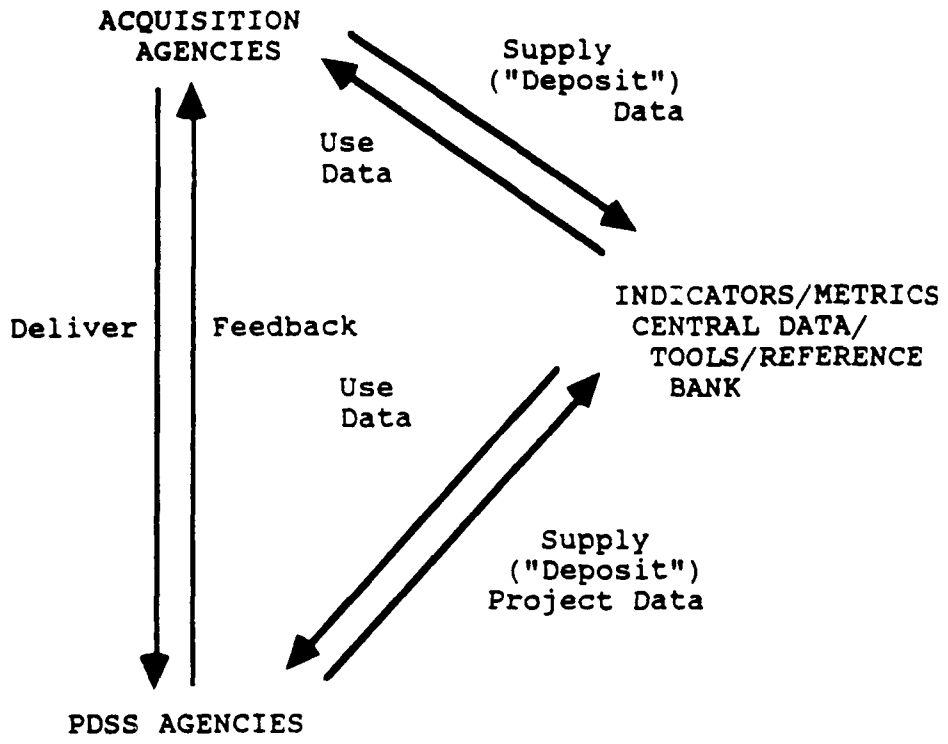


FIGURE 5. Collection of PDSS Data

Collection of quality metrics data on fielded systems of various types (languages, applications, and other meaningful parameters) and storage in the central data bank will provide an early baseline to assist in establishing and refining evaluation criteria. This will assist in maturing the quality metrics at a rapid rate. This effort of collection should use existing automated tools to the extent possible and all newly developed tools should be put into the central data bank for future use.

IMPLEMENTATION.

Initially a multiservice funded effort, with a lead Service in control.

BENEFITS.

This data base will assist program management in determining acceptance criteria on future programs and assist PDSS in establishing resource requirements. This will also serve to validate the tools in use and refine their use for estimating purposes. In summary, the major benefits of the data base are its low cost and high potential for long term payoff.

(Intentionally Blank)

HUMAN RESOURCES IN PDSS
PANEL VI
PROCEEDINGS

OBJECTIVE.

The primary objective of the Human Resources Panel was to define actions necessary to enhance the recruitment, retention and training of knowledgeable software personnel to support PDSS.

The basis for the establishment of Panel VI grew out of the Orlando I JLC Software Workshop discussions. Since human resource issues are an integral element impacting all other panel issues it was determined that a separate panel was required for Orlando II to assess and recommend improvements in the software personnel arena.

SCOPE.

Approach. The panel consisted of 14 members and was comprised predominantly of government civilian representatives. As a result, the discussions focused on the concerns and needs impacting the federal civilian software community. Panel members had been informed of panel issues and the primary thrust of the charter by letter from the co-chairmen. Extensive reading materials were forwarded prior to the workshop to provide needed background information in order to facilitate discussions. Additional reference material was solicited for use by the panel. Following the workshop opening session, the panel convened and discussed organization, schedule, and the panel charter.

The charter objective was defined by the panel under three broad topic areas:

1. Improving the career management structure for software personnel.

2. Promoting continuing education and training programs to enhance the knowledge and skills of the existing software work force.

3. Promoting expansion of university/college curricula to include courses in new software technology and modern engineering practices.

The panel then divided these broader issues into the following subgroup task break-outs:

Subgroup 1 - Career Management Issues.

1. Recruitment
   a. Availability of qualified candidates
   b. Mechanisms

2. Retention
   a. Personnel turnover
   b. Professional recognition
   c. Pay incentives

3. Career Patterns
   a. Work force structure
   b. Skills classification
   c. Career progression

4. Management Initiatives

5. Enhanced Utilization

Subgroup 2 - Education/Training.

1. Educational Programs
   a. Curricula development
   b. Resources

2. Training
   a. Programs and requirements
   b. Tools, techniques and aids
   c. Define costs

The two subgroups, which convened on Tuesday through Thursday, researched the respective issues and prepared the discussion, conclusions, and recommendations that follow. Summary interim reports of panel results were presented to the entire workshop.

DISCUSSIONS/ISSUES.

Introduction. The panel recognized very early in its discussion that the human resources objective was too broad in scope and that the DOD personnel situation is a complex and multifaceted area which includes people, organizations and regulations. The panel reviewed the "STARS Functional Task Area Strategy for Human Resources" report, published by DOD in 1983, which identified six major subtask areas related to personnel and education. The human resources panel was not in a position to tackle a detailed analysis of all these subtask areas identified in the STARS report and decided to focus their attention on more immediate problems and concerns, such as those outlined in the Air Force BOLD STROKE Action Plan.

Project BOLD STROKE detailed four objectives to attack software problems:

1. Awareness
2. Education and Training
3. Personnel Management
4. Future Planning.

The thrust of such initiatives coincided with the discussions and recommendations developed by the human resources panel.

External Constraints. Personnel policies designed for software personnel are subject to numerous external pressures and constraints. The issue of human resources has been addressed from the standpoint of improving the ability of the government to attract and retain knowledgeable software engineers and to maximize their proficiency through proper training and incentive programs. The authority and financial resources necessary to build an adequate staff has been assumed as given. It is recognized that the trend is toward heightened austerity and that the expectations may not be realized. Past efforts to justify resource requirements have been largely unsuccessful, mainly because the magnitude of the estimates has been disturbingly high, and the software support community has not been successful in gaining credibility for its estimation technique.

Further, the impact of inadequate resources on operational readiness of the Mission Critical Defense Systems (MCDS) has not been convincingly portrayed to the decision makers for several reasons. First, it is impossible to forecast what kinds of failures will occur and to what extent they will degrade the systems' capabilities. Secondly, as each year goes by with the software support organizations resourced to only a fraction of the stated requirements, there is no noticeable short range consequence. Additionally, the long range implications of continued inadequate staffing are too ephemeral to gain support in an environment of scarcity. Even the argument that the workload is increasing because the number of MCDS is increasing has been unsuccessful.

While the efforts to acquire additional resources must not abate, it is clear that an aggressive marketing and education program is necessary to achieve even as modest a goal as relief from anticipated manpower reductions.

MANPOWER SHORTAGES/RESTRICTIONS ISSUE.

In a time of steadily decreasing manpower ceilings, it is unrealistic to seek higher authorizations. Therefore, it is increasingly important to seek to protect the existing authorizations from further erosion. Job series which require

critical skills tend to be the series which experience the greatest turnover. The normal technique for implementing reductions in authorized slots is through attrition by eliminating vacant positions first. Were that to continue, the software support activities, because of the higher than average turnover would be unfairly penalized. Such reductions would be burdensome with a stable workload; with an expanding workload it is intolerable.

Proposed Solution. Protect the critical software engineering skills from further cuts by "fencing off" the existing spaces.

Projected Benefits.

1. Reduced reliance on costly contractual support.

2. Retention of an in-house cadre for the preservation of corporate memory and maintaining technical expertise.

3. Retention of trained, knowledgeable employees to avoid disruption of on-going system support.

Final Recommendation (Recommendation 4-6-01). The JLC actively promote the exemption of positions for software support of MCDS from manpower reduction actions.

CAREER MANAGEMENT ISSUES.

The availability of a highly skilled computer and software engineering work force to support MCCR requirements is a vexing problem and will continue to impact how the PDSS effort will be staffed. Technical requirements should drive the PDSS staffing mix, and the mix should be made available through proper planning and implementation. PDSS is, and will continue in the near future to be, a labor intensive activity. The demand for software by DOD is anticipated to increase at a rate of 12 percent per year for the next two decades, according to the EIA. Currently all services are building PDSS staffs using primarily electronic engineers (GS-855), computer scientists (GS-1550), computer specialists (GS-334), and to a small degree mathematicians (GS-1520). There are two major sources from which activities recruit for these skills:

a) Recent college/university graduates for entry level positions GS-5/7.

b) Experienced engineers and professionals.

The DOD has been faced with the constant challenge of recruiting sufficient numbers of engineers to meet its growing mission needs, especially in the MCCR area. DOD agencies recognized that

they needed to address this problem through the development of specialized and cost effective recruiting techniques, as well as designing innovative programs that would provide alternate sources of trained PDSS personnel. The use of direct hire authority, special salary rates, accelerated training plans, payment of pre-employment interviews and relocation costs to first duty station for critical skills occupations has greatly enhanced DOD's ability to attract entry level civilian engineers.

Innovative recruitment initiatives, such as the joint AFLC and University of Dayton Re-entry Program, is just one example of programs implemented within DOD activities to provide additional sources of engineering talent to support MCCR requirements.

1986-87 private industry hiring reductions have also improved DOD's recruitment posture, though agencies are still experiencing difficulties in attracting experienced professionals especially for certain geographic locations. DOD work force trends indicate that attrition rates have dropped significantly since 1984. Turnover rates for computer professionals, particularly software engineers, still tend to be higher than other occupations. Computer professionals and software engineers are a multi-industry resource, and therefore have a variety of alternative opportunities. Their jobs are not extremely sensitive to supply and demand forces within particular industries, because if one industry is in a slump, computer skills can often be transferred to another one that is prospering.

Issue 1. The panel noted that DOD is not positioned to be competitive in recruiting and retaining experienced PDSS personnel, limiting our ability to meet the existing and projected software engineering requirements. The panel agreed that current personnel systems are cumbersome and that government agencies need greater flexibilities in assigning rates of basic pay in order to recruit, motivate and retain a well qualified work force. New career management procedures are required.

Proposed Solution. Pay banding concepts, which are an alternative or simplification of existing position classification and pay systems, have been implemented within the DOD through various demonstration projects. This approach has been incorporated into a DOD legislative proposal entitled "Civil Service Simplification Act of 1986". Such legislation would allow the Naval Demonstration Project (i.e. pay banding concepts) to be incrementally expanded throughout the federal work force in a controlled, measured and budget-neutral manner. Other benefits of the DOD legislation is that it ties pay and retention to performance and is open to any occupation, activity or geographic area. Also included are changes to special salary rate provisions which would allow for special rates in a greater variety of circumstances, increase the available rate range when necessary and permit the hiring of individuals covered by special

salary rates at a rate above the minimum established for that special rate range. The proposal would also permit the payment of recruitment or retention bonuses based on continued service agreements.

Final Recommendation (Recommendation 4-6-02). Although up until now, available avenues have enhanced the government's ability to hire entry level scientists and engineers, this will not always be the case. We therefore recommend the JLC endorse, through appropriate channels, the proposed DOD legislation reiterating the need for greater flexibility in rewarding the efforts of our senior level PDSS personnel.

Issue 2. The panel discussions also surfaced the problem that the federal government is unable to assign civilian personnel having requisite skills in computer and software engineering to appropriately classified and structured positions. A triservice initiative, chaired by the Navy, proposed a new classification standard covering computer engineering (GS-8XX) and a revision of the computer scientist (GS-1550) series. These new standards were submitted to OPM for review and approval. DOD is expecting OPM to officially release this new computer engineering standard shortly. In conjunction with these new classification standards, OPM should take steps to revise the X-118 Qualification Standards to incorporate the computer engineering series. Current qualification standards do not address electronic and software engineering course work under their basic requirements.

Proposed Solution. Amend the X-118 qualification standards to include the computer engineering series. Modify the basic requirements by inserting additional course areas relevant to electronics and software.

Final Recommendation (Recommendation 4-6-03). We recommend that JLC request appropriate revisions to the OPM X-118 qualification standards to incorporate computer engineering or software engineering course work and reflect the new technologies.

EDUCATION AND TRAINING ISSUES.

Improving the productivity of software engineers requires new ways of thinking and reasoning about software and better methods of producing it. To gain intellectual control over the software production process and become more productive and efficient, the DOD is aspiring to make the production of software less labor intensive and more technology intensive. The use of these new technologies requires users to be better educated and trained.

Panel discussions began by defining education and training. Education is a long term activity based on fundamentals, and

designed to build a foundation of knowledge and reasoning abilities. The panel agreed that education fell into two categories:

1. Initial development of skills required to begin a software engineering career (i.e. Bachelor's Degree).

2. Continuing education requirements to keep abreast of advancing technologies.

Training is a short term activity with a specific goal, and builds upon the educational foundation. The challenge to educators is to provide the appropriate foundation for software engineers, so that the expected rapid advancements in technology can be used effectively after relatively short training periods.

Issue 1. There are currently very few undergraduate curricula which provide the PDSS community with entry level professional software engineers. To increase the number of qualified software engineers we need to increase the number of software engineering educational programs. To achieve this goal, we must work against the enormous inertia of an education system that does not respond quickly to new educational needs.

Proposed Solution. In September 1983, the Educational Activities Board of the IEEE Computer Society published a model curriculum program in computer science and engineering which addressed curricula and guidelines for the development of facility, administration and material resources.

The primary goals of the model program were.

1. To provide and define curricula features of undergraduate programs in computer science and engineering.

2. Provide a standard of comparison that could be used to guide the development of new programs or the modification and upgrading of established programs.

3. Provide standards for ABET accreditation.

4. Provide guidance to academic administrators concerning the level of commitment needed to support a program.

The DOD contract that established the SEI specifically mentions education as a mission of the Institute, saying:

> "It shall also influence software engineering education curricula development throughout the education community."

Its proper role is to serve as a focal point and catalyst to influence software engineering curricula. Through its Education Division, the SEI should take the lead in marketing the IEEE model program in computer science and engineering to respective colleges/universities in order to increase the number of new software engineers.

Projected Benefits. An undergraduate software engineering curricula would provide for a work force that is better equipped academically to support PDSS and be prepared for the anticipated transition to a technology intensive activity in the next decades.

Final Recommendation (Recommendation 4-6-04). That the JLC refine and market through SEI, a model for computer/software engineering curriculum, such as the IEEE proposal, which would enable colleges and universities to readily implement and expand software engineering programs to adequately prepare students for such professions.

Issue 2. There currently exists in the DOD community a critical need for a consolidated and concise approach to software engineering training and an increase of awareness at the middle management level of the need for such training. The training of our software engineers basically falls into two categories. The first of these is the continuing education/training of those individuals already working in the software area. The second and perhaps the more critical is the problem of cross-training individuals from other technical disciplines into that of software engineering.

There are several problems facing the DOD community relative to meeting these training needs. In many instances, middle level managers are not even aware of the impending need to train their people to meet the PDSS challenge. In those organizations where the need is recognized, the manager is unaware of the numerous training courses already in existence. Often times, new training courses are developed "on the fly" duplicating those already in existence only to find out after-the-fact that the course has missed the mark relative to meeting their specific needs. The cross-training problem is even more acute in that the manager often times does not have a structured mechanism for identifying and selecting those individuals or courses available to meet the larger task of career field changes. Without solutions to these problems, the projected DOD software personnel shortfall will remain unanswered and the PDSS challenge will not be met.

Proposed Solution. There are several solutions to the problems:

1. Create a DOD wide mandatory training program to educate and raise the awareness level of DOD middle level managers. The AF project BOLD STROKE is a first step in this direction.

2. Create a data base of all current DOD and commercial software training courses. This data base should be implemented on electronic media and as a minimum should consist of an abstract describing the contents of the course.

3. Investigate a mechanism of updating and making available to the DOD government and industrial community easy access to this data base.

4. Develop a mechanism to mandate cross-training of selected DOD personnel to the software engineering career field.

5. Establish guidelines and procedures for selecting those individuals for software cross-training programs.

6. Ensure training funds are available to meet the mission critical PDSS software training requirements.

Final Recommendations (Recommendation 4-6-05). Charter a Joint Service/Industry task ad hoc group to assess PDSS training courses and service needs, define a consolidated approach to software engineering training, create awareness in DOD management of software training & funding requirements and develop an automated training data base.

SUMMARY OF RECOMMENDATIONS.

1. Establish a new software engineering job series (GS-8XX) for the civilian work force and request revision of OPM X-118 Qualification Standards for professional engineering series.

2. Adopt alternative position classification and pay systems (i.e. "pay banding") by supporting DOD legislative proposal - Civil Service Simplification Action of 1986.

3. Refine and market a model for computer engineering/ software engineering curriculum.

4. Task an ad hoc group to:

a. Define a consolidated approach to software engineering training.

b. Create awareness in DOD management of software training and funding requirements.

c. Assess available training and service needs.

d. Develop an automated data base.

5. Protect existing manning levels by "fencing off" critical PDSS spaces.

(Intentionally Blank)

PANEL VI BIBLIOGRAPHY.

1. ACM81 - Committee on Computer Curricula of the ACM Education Board, "ACM Recommended Curricula for Computer Science and Information Processing Programs in Colleges and Universities, 1968 - 1981", ACM, New York, 1981.

2. Boehm, B., Penedo, M., Stukle, D., Williams, R., "A Software Development Environment for Improving Productivity", IEEE Computer, June 1984, pp. 30-42.

3. Redwine, Samuel T., et. al., "DOD Related Software Technology Requirements, Practices, and Prospects for the Future", Institute for Defense Analysis, Alexandria VA, IDA Paper Number P-1788, June 1984, pp. 1-139.

4. "Report of the DOD Joint Service Task Force on Software Problems", Deputy Under Secretary of Defense for Research and Advanced Technology, Pentagon, Washington DC, July 1982.

5. "The Professional Environment in Army Laboratories and Its Effect on Scientific and Engineering Performance", prepared by Committee on Army Manpower, Board on Army Science and Technology, Commission on Engineering and Technical Systems and National Research Council; National Academy Press, Washington, DC 1983.

6. SEI, "Graduate Curriculum for Software Engineering Education", Carnegie-Mellon University, Pittsburgh, PA, April 1986.

7. Gibbs, Norman E., Ford, Gary A., "The Challenges of Educating the Next Generation of Software Engineers", SEI-86-TM-7, Carnegie-Mellon, Pittsburgh, PA, University, June 1986.

8. "Model Program in Computer Science and Engineering", Draft 1983, IEEE Computer Society.

9. "Software Technology for Adaptable Reliable Systems (STARS) Functional Task Area Strategy for Human Resources", Department of Defense, March 1983.

(Intentionally Blank)

SOFTWARE TECHNOLOGY TRANSITION
PANEL VII
PROCEEDINGS

OBJECTIVE.

The stated objective of the panel was to identify policies and practices for transitioning necessary tools and methods while controlling their proliferation, so that PDSS needs are met in a cost effective manner. In connection with this objective, two panel tasks were identified:

1. Identify problems and recommend solutions for the insertion of support tools and new technologies into PDSS activities.

2. Identify problems and recommend solutions for the transition of operational software (tactical programs) from the developing to the supporting organizations.

BACKGROUND.

PDSS and, specifically, technology transition has been addressed with varying levels of interest and intensity over the last ten years. While there has been some progress with respect to PDSS policy, e.g. OPNAVINST 5200.28 and DOD-STD-1467, the issue of technology transition has been stalemated. Major initiatives in this area have been oriented more towards development activities than support activities. They include the STARS and the SEI.

The STARS program contained numerous technology transition thrusts and, while still a viable DOD program, has been undergoing serious realignment, principally focused on the introduction of Ada. The SEI has been established with a major project in technology transition methods. This program, however, is directed towards understanding the process, as opposed to near term "injection" projects. PDSS technology injection is a current and dynamic problem as the three Services and the Marine Corps are supporting hundreds of MCDS.

SCOPE.

The panel considered all factors that influence the transition process as they impact PDSS: DOD policy, contractor posture, contractual issues, specific technologies (tools/methods) and other practices such as configuration/data management. Included in the analysis was the investigation of the development phases of the system life cycle, its impact upon the support phase, and injection of technology directly into PDSS.

ASSUMPTIONS AND CONSTRAINTS.

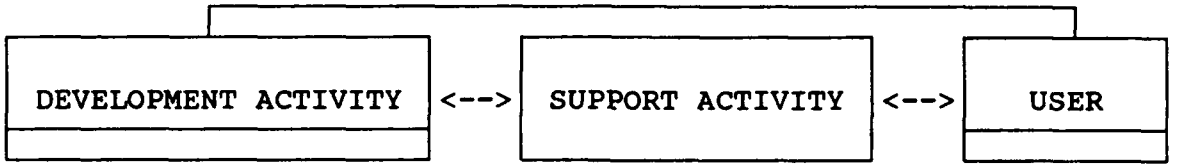
The panel concentrated on near term solutions/recommendations in order to impact current situations. In addition, the panel considered on-going programs such as STARS/Ada and thrusts such as the SEI. The intent was purposely not to conflict with or duplicate DOD, Service, and command policies and practices.

The panel was cautious not to propose recommendations that were unwieldy, e.g. create new Service organizations. At the same time, if major recommendations were deemed necessary, the panel did not shy away from them.

It was recognized that no standard environment/tool set currently exists, even though there have been thrusts in that direction. While this goal was not attacked, the panel felt that realism precluded one standard environment, therefore, solutions have been oriented towards a heterogeneous environment.

APPROACH.

Preparation. A preliminary meeting was held on 9 December 1986, to discuss the purpose of the panel, its scope and specific objectives. The following model was proposed:



Contractor Activity Support

Contractual Issues

Government Activity

Policies/Practices

PDSS Environment

Policies/Practices

Customer

Each of these areas has policies and practices that affect the supportability of systems. The objective of the panel was to identify these policies/practices in order to understand their impact and suggest improvements. For example, the tools that are used in development (by a contractor) can be different from the tools required by the support activity.

In general, the panel used an approach using informational briefings and brainstorming to formulate issues/problem areas, then breaking into smaller groups to attack specific issues areas, and finally, developing a general panel consensus on recommendations.

Issue Identification. A suggested work flow was developed at the 9 December 1986 panel meeting that was used to develop the points and issues for the subpanel reports. At this meeting four issue areas were identified:

1. Contractual Issues
2. DOD Policies/Practices
3. Software Tools and Environments
4. Impact of Ada

During the panel briefings and discussions, additional subissues were identified for considerations. Furthermore, based upon the panel deliberations, the fourth area, Impact of Ada, was folded into the other three areas. Three subpanels were created and Subpanel Chairmen were designated to evaluate each area:

1. Mac Murray (Chair) - Contractual Issues
2. Perry Nuhn (Chair) - DOD Policies/Practices
3. Jack Cooper (Chair) - Software Tools and Environments

SUBPANEL ISSUES.

The resulting issues and subissues are summarized in the following paragraphs.

Subpanel 1 - Contractual Issues.

1. Data Rights -- Impact on tools and technologies.
2. Current contractor practice and its effect on transition.
3. Contract Unique requirements.
4. Contractual Instruments -- Firm Fixed Price versus Time & Materials.
5. Acquisition of Tools for Operation & Maintenance.
6. Use of Government Furnished Equipment/Government Furnished Property.
7. Software acceptance criteria.
8. Enforcement/usage by contractors of tools/methods/ procedures/software engineering.

Subpanel 2 - DOD Policies/Practices.

1. Current Government policy/practice and its effect on transition (including NDI, HOL policy, etc.).

2. Current Government support posture (e.g. PDSS) and its effect on transition.

3. Organizational structure of Development & Support Agencies.

4. Sources of Funding (R&D, OMA, ...).

5. Acquisition of Ada Tools and Software Engineering Environments.

6. PDSS Transition Starts too late (OPNAVINST 5000.28).

7. Standardization Requirements (DOD-STD-2167, DOD-STD-2168).

8. Transition Practices (Transition Plan).

9. Start-up/Transition Penalties.

10. Nonstandard Acquisition Strategies.

Subpanel 3 - Software Tools and Environments.

1. Methods of speeding up the transition and infusion of state-of-the-art tools into PDSS environments.

2. Differences between Acquisition and O&M and resulting impact on the transition of tools/methods into the PDSS environment.

3. Problems with the transition of operational software (tactical programs) and identification of methods to streamline the transition process.

4. Influence of software environments (APSE, etc.) on the transition of tools.

5. Current Service support postures (e.g. PDSS) and their effect on transition.

6. Transition of SSE and tools will be evaluated from the supporter's point-of-view.

7. Integration and interoperability of Tools & Environments.

PANEL PROCEEDINGS.

General. The first day's meeting was used for orientation, resolution of issue areas, assignment of personnel, selection of subpanel chairmen, etc. Since the workshop plan was to complete a written draft report by the end of the workshop, maximum time was devoted towards subpanel meetings.

The specific approach included morning sessions for information briefings and cross-communications and afternoon subpanel working (writing) sessions. The purpose of the informational briefings was to create a common level of knowledge across panel members. The informational briefings were scheduled throughout the panel's deliberations, based on their need at a particular time.

INFORMATIONAL BRIEFINGS, SUMMARY.

The following paragraphs summarize the informational briefings that were presented during the panel meetings:

1. Service Practices on PDSS, Army, Navy. Jack Cooper, Anchor Software Management, provided an overview of Army PDSS responsibilities and functions of the Army's Centers for Life Cycle Software Engineering (CLCSE). He noted that although the Army does not have an unified policy for software development and life cycle software support, subordinate commands have developed their own regulations and policies. Also discussed were the role of the Computer Resource Management Plans (CRMP) and difficulties in enforcing standards and policies. Several contrasts were noted between Army and Navy policies and procedures.

2. DOD-STD-2167(A) Proposals. Capt Richard Schmidt, Air Force, discussed software development practices and tailoring requirements when using DOD-STD-2167, Defense System Software Development. He noted industry objections to DOD-STD-2167 and discussed proposed government resolutions to the objections.

3. NAVAIRSYSCOM Policy. Tom Smith, NAVAIRSYSCOM, presented a briefing on NAVAIR Instructions, "Tactical Embedded Computer Resources (TECR) Policy in the Naval Air Systems Command", and discussed aspects of NAVAIR experience in developing software policies. He then highlighted concepts of NAVAIRINST 5230.9, the establishment and operation of software support activities. He stressed the importance of organizational structures, coordination with various field activities, and Navy policies on standardization.

4. Ada in Mission Critical Computer Systems. Wolfhart Goethert, IIT Research Institute (IITRI), presented a short briefing on IITRI's recent study of software engineering technologies. The study concluded that although substantial Ada

implementation is planned and, in some cases initiated, the PDSS impact of Ada will not be felt until the late 1990's.

5. DOD-STD-1467. Chuck Gordon of CACI, Inc., presented an overview of DOD-STD-1467, Military Standard Software Support Environment, as applied to the Army. The objective of this standard is to establish a complete life cycle support capability. The standard attempts to ensure compatibility between the development and support environments.

6. PDSS Data Rights. Anne Martin, of the Software Licensing Project of the SEI, gave a presentation on the SEI's recent study of data rights issues in software life cycle support. SEI concluded that since PDSS requires a transition of system expertise from the developer to support personnel, conflicts arise between DOD's needs and industry's proprietary rights. This requires flexible software acquisition policies, which consider the needs of DOD with the proprietary rights of industry.

7. Software Engineering Environments. Hank Stuebing, Naval Air Development Center, presented an overview of the Navy's experience with the STARS Software Engineering Environment (SEE) development. He explained that nearly a dozen architectural studies have been completed in efforts to ease the evolution of methods, tools, and procedures. He cautioned against oversimplifying solutions of information interface exchanges.

8. Service Practices on PDSS, Air Force. John Marciniak, Marciniak and Associates, provided an informal discussion on the Air Force perspective on PDSS and supportability requirements.

DISCUSSION - PANEL ISSUES/RECOMMENDATIONS.

The products and recommendations that resulted during the panel deliberations were resolved in panel planning sessions to produce a consensus report. The three perspectives/issue areas used to address technology transition, and the eight recommendations that resulted from these three areas are listed below. The eight recommendations are discussed in the following section.

1. DOD Policies/Practices Issue Area.

   Promulgate DOD PDSS policy.
   Promulgate DOD software support policy.
   Improve PDSS training for managers.
   Promulgate software support environment standards.

2. Contractual Issue Area.

   Improve acquisition regulation support.

3. Software Tools and Environments Issue Area.

Establish PDSS software commonality office.
Modernize tools and technology for PDSS of pre-Ada.
Systems develop Ada conversion criteria.

DOD POLICIES/PRACTICES ISSUES/RECOMMENDATIONS.

DOD PDSS POLICY.

DOD level policy is needed to explicitly address PDSS within the system development life cycle. Current overall system acquisition policy is less than adequate and seriously outdated. This policy (DODD 5000.29) and its implementing instructions provide guidance to DOD and should be maintained in an up-to-date status.

Recommendation 4-7-01. Specific actions which should be undertaken include:

- Develop and issue a DOD Instruction to implement the policy described in the current version of DODD 5000.29.
- Strengthen the OSD oversight function (individual) for cognizance over software development and support decisions.
- Update and re-issue DODD 5000.29 to emphasize PDSS consideration during the acquisition process.

This is considered as a near term action with significant immediate and long term ROI. Policy development and promulgation is important to ensure adequate logistics support. OSD should establish close liaison with PDSS support elements within each Service/agency.

Near Term (0-1 Year) Solution.

- Develop and issue a DOD instruction, implementing current policy contained in DODD 5000.29.
- Strengthen the OSD oversight function.
- Establish close liaison between OSD and PDSS support elements within each Service/agency.

Mid Term (2-4 years) Solution.

- Update and reissue DODD 5000.29 to emphasize PDSS.

Return on Investment. Each of these actions are considered to yield significant immediate and long term ROI. Cost and time to implement include staff resource and schedule requirements.

Method of Implementation. The JLC should communicate problem/ recommended solution to cognizant OSD executive. Formal communication and/or briefing by the JLC is suggested.

Justification of Priority. Policy establishment and promulgation is the key to obtaining visibility/achieving implementation. This action is considered to be relatively low cost and easy to implement with very high ROI.

DOD SOFTWARE SUPPORT POLICY.

There is currently no uniform DOD policy used in contracting for support software. Most contracts only address requirements for the acquisition of operational software. These contracts typically fail to specify software requirements related to life cycle support. The Army has recently developed and promulgated a Military Standard specifically for this purpose. This standard is a positive step in the right direction and offers significant potential for broader DOD usage.

Recommendation 4-7-02. DOD-STD-1467, SSE, dated 18 January 1985, should be reviewed, modified (if applicable), approved, and promulgated on a DOD wide basis. The review of DOD-STD-1467 should also consider the inclusion of PDSS technology transition requirements that must be fulfilled to prepare the PDSS for system turnover, acceptance, and support. DOD (OSD) and the Services should establish a comprehensive cross-Service review of DOD-STD-1467 and promulgate its extended use throughout the DOD.

Return on Investment. ROI is medium to long term. The estimated cost is unknown, but considered to be within current Service staffing levels.

Alternatives. The alternative is to promulgate individual Service standards. This alternative is not recommended.

Justification. The priority of the task is justified in that:

- It provides requirements in the system acquisition life cycle.
- It better ensures that managed transition of the system and the related technologies necessary to support it throughout its operational life.

PDSS TRAINING FOR MANAGERS.

DOD and contractor PMs who develop systems with PDSS requirements do not thoroughly understand the software development process, the software life cycle, and the impact of supportability issues on the final products.

Recommendation 4-7-03. DOD and contractor PMs who develop systems with PDSS requirements need an understanding of PDSS issues to adequately plan and execute pre-PDSS activities. Therefore, PDSS training will be necessary for PMs not thoroughly familiar with PDSS and related technologies. A three level training program is proposed: (1) short video tape, (2) one day tutorial and (3) two day tutorial.

Term of Solution. A short video tape based tutorial, with wide dissemination, is the near term solution. This will, in effect, "get the word out." The tutorial is also part of the long term solution for high level PMs.

Return on Investment. ROI is very high. Estimated costs and implementation time are:

- Video Tape: $50K and 1 year.
- Short Tutorial: $100K and 2 years, plus $1K/student administration cost.
- Long Tutorial: $150K and 2 years, plus $1K/student administration cost.

Dependencies. To be incorporated with existing PM Training.

Method of Implementation. Already illustrated; video tape presentation, short tutorial, long tutorial.

Justification. Technology advancement, system development, and PDSS are complex interactive issues. PDSS can be detrimentally impacted from decisions made during system development. Informed PMs are necessary for successful PDSS planning and execution.

Detailed Product. The short video tape is intended for high level PMs. It should include (not in any order):

- PDSS crisis.
- Appropriate standards and regulations.
- Technology trends.
- All Services' approach to PDSS.
- View from DOD and the Services.
- View from the support organizations.

The in-class tutorials are intended for PM, Chief Engineer, and lower level management personnel. They should include an in-depth discussion of topics included in the video tape. Tutorial presenters should be experienced PDSS personnel.

SOFTWARE SUPPORT ENVIRONMENT STANDARDS (CAIS Implementation).

There is a proliferation of software support environments and like tools within these environments throughout DOD. Service direction and policy should be developed and effected as soon as possible in order to establish the DOD support base to encourage rapid implementation by contractors and agencies.

Recommendation 4-7-04. DOD-STD-1838, CAIS, is due for printing and distribution in February, 1987. This version will provide host and operating system transportability of tools used in PDSS activities. It will address the multiple problems related to specific hardware/operating systems used at PDSS organizations. It will enhance technology transition by providing standard interfaces to plug tools into PDSS support environments, thus allowing easy use of new tools on existing or new hardware and/or operating system.

Near Term. In the near term (0-1 year), productivity of the PDSS organizations would not be significantly improved while tools are being developed that conform to the CAIS standard.

Mid Term. During the period 2-4 years after implementation, the productivity of the PDSS organizations will steadily increase while resources (systems and operating systems) will begin to stabilize. This positive reaction is the result of tools becoming processors/operating system independent through the effective implementation of CAIS.

Long Term. The long term benefits will further increase productivity and stabilize resources due to the delivered products becoming tool independent. The PDSS Software Commonality Office should be tasked to be the Service advocate/clearing house for CAIS implementation activities.

Implementation. Cost for implementation of CAIS includes continuing tool performance penalties, tool conversion costs, and CAIS shell implementation and certification.

CONTRACTUAL ISSUES/RECOMMENDATIONS.

ACQUISITION REGULATION SUPPORT.

The JLC needs to support the DOD Acquisition Regulations which include the DAR, FAR, and DOD FAR Supplement. There are two concerns with the current DOD data rights policy:

1. Contractors are unwilling to utilize their most sophisticated tools or development efforts if they may have to deliver those tools, with unlimited rights, to the government;

2. Entrepreneurial companies are unwilling to do business with the DOD for fear of losing competitive advantage.

It is not clear that DOD data rights policy would require the transition of tools, or that it would rob small firms of their competitive advantage. However, a recent SEI survey has established that the fear of these conditions has caused companies to not do business with the DOD or to restrict their use of tools on DOD products.

Recommendation 4-7-05. It is necessary to develop or to adjust the Acquisition Regulations so that state-of-the-art tools are available for PDSS. This may require selectively requesting unlimited rights on procurements; it may require that PDSS facilities have the option to mandate that deliverable software be supportable by existing or commercially available tools. Once Acquisition Regulations that encourage the transition of technology into the PDSS environment have been developed, the policy must be communicated to PMs, to the PDSS community, and to DOD contractors.

Near Term. Near-term developments are centered at the SEI's Software Licensing Project (SLP). SEI should develop a summary of their Acquisition Regulation work and disseminate it widely.

Mid Term. The policy will be progressing through the approval cycle, and information on the policy will need to be disseminated to PMs, contractors, and PDSS centers.

Long Term. The Acquisition Regulations will be approved and a handbook on their use developed.

Return on Investment. The advantage of this policy will be the transition of support tools to PDSS centers in the short term and the development of a de facto standard and clones in the long term. System-specific support tools will gradually disappear.

Dependencies. The acceptance of DOD-STD-1467 and the Acquisition Regulations will be complementary in the evolution of a consistent, integrated PDSS support environment.

Alternatives. Either the Acquisition Regulations must have a data rights clause that encourages transition of state-of-the-art tools or the restrictive clause must be ignored. The lawyer-out-of-the-loop approach is, unfortunately, not going to happen.

Implementation. The SLP of the SEI is developing Acquisition Regulations which will eventually be implemented. The SEI would be the first choice for communicating the ramifications of the policy to PMs, PDSS personnel, and government contractors.

Justification. Current Acquisition Regulations, or the fear of potential losses due to them, are prohibiting the transition of state-of-the-art PDSS tools. The current Acquisition Regulations do not support mandating that delivered software be supportable by an existing PDSS environment.

Product. The products are Acquisition Regulations and the handbook to tailor them.

SOFTWARE TOOLS AND ENVIRONMENTS ISSUES/RECOMMENDATIONS.

PDSS SOFTWARE COMMONALITY OFFICE.

The separation of the Services, the organizational and command separation within each Service, the alignment of PDSS organizations along acquisition program lines, and concentration on immediate operational problems inhibit the identification, procurement, and widespread distribution of common PDSS tools, methods, and processes.

Recommendation 4-7-06. Each Service should establish at the command level a PDSS software commonality office with the following charter:

- Identify, evaluate, procure, and distribute tools and methods to users at PDSS activities.
- Provide centralized support for user assistance, consolidation of user requirements, and resolution of software problems.
- Provide coordination between the Services and raise the level of visibility of PDSS concerns.

Term of Solution. Each office can be established in less than a year and should consist of a project engineer, a staff engineer, and a budget analyst. Funding will be required for tool procurement, tailoring, and maintenance.

Return on Investment. The estimated ROI will average 10 to 1 by eliminating duplicate efforts at PDSS activities.

MODERNIZATION OF TOOLS/TECHNOLOGY FOR PDSS OF PRE-ADA SYSTEMS.

PDSS requirements will increase significantly, without a commensurate increase in resources: the software crisis. DOD is attacking this problem for future systems with Ada technology; however, a majority of systems supported by PDSS for the next 15-20 years will not be in Ada. As a result, Ada productivity improvements will not be realized by PDSS activities for some time, and, potentially, PDSS will not have the resources for adequate support during this transition period.

Recommendation 4-7-07. The rapid implementation of the Ada language must continue. However, technology transition must also be initiated for the systems that are currently being developed, being fielded, or that are already deployed. DOD (OSD), in addition to aggressively pursuing the Ada technology efforts, should also pursue an other-than-Ada software engineering technology improvement program for PDSS technology improvement. The program should include increased tasking to the STARS program and the SEI.

Term of Solution. The solution is mid and long term.

Return on Investment. The cost is unknown, but should require only minimal increases in current funding for STARS and SEI. For example, current STARS "target example environments" could address the PDSS need.

Justification. Prioritization is justified on the basis that:

- It provides for continued technology improvement for the support of existing and forthcoming weapons systems which are non-Ada based.

- It may provide the sole way that these can continue to be supported on a cost-effective basis.

Implementation. The method should be determined by OSD.

Product. The end product will be the development and distribution of software technology that will increase the productivity of PDSS activities for Ada systems and other than Ada systems.

ADA CONVERSION CRITERIA.

It is generally recognized that systems implemented in Ada will be much easier and, hence, less costly to maintain. However, because Ada is only now being required for developing systems, there are large inventories of software in existence that are not in Ada, but which will have to be supported for the next 20 to 30 years. It is estimated that by the year 2000, this pre-Ada software may still represent up to 80% of supported inventory. The reduction of this pre-Ada inventory is clearly desirable. However, it is not easy to determine when and if a given system should be considered for upgrade to Ada.

Recommendation 4-7-08. To allow conversion decisions to be made in a logical manner, it is necessary that criteria be established that will allow cost effectiveness to be established for the upgrade alternative. SEI should be tasked to collect and analyze data indicating the investment needed to upgrade various types of

systems to Ada, and the expected payoff in increased maintenance efficiency that would result.

Term of Solution. It is anticipated that a study to establish criteria could provide useful results within six months. Therefore, this is a near term solution that will pay off over the next 20 years.

Return on Investment. It is estimated that such a study would require approximately 2 to 3 personnel over a six month period. The direct payback of the results over the next 20 years can easily be on the order of 100 to 1.

Implementation. This action should be implemented by tasking SEI to conduct the study with the developed criteria provided to the Services for utilization.

Justification. This study is considered a priority action because it has the potential to significantly impact the PDSS burden of pre-Ada software.

Product. The product should be a report that analyzes the nature, size and level of documentation of various systems, the expected cost of conversion to Ada, and the payoff in terms of anticipated future maintenance activity. Development of a computer program/model should also be considered.

CONCLUSIONS.

In recent years, advances in automating the software development process have been focused, almost exclusively, on the development of tools to speed up the design and coding process. While there have been activities directed at PDSS improvement, e.g., the Army's DOD-STD-1467, and the organization and establishment of PDSS activities, i.e., the CECOM CLCSE, the Navy's facilities for the F-14, etc., and the Air Force's F-111, predominant attention has been directed at software development issues. There is a dire need to direct a PDSS program, in terms of both policy and technology, to improve technology transition into the PDSS environment. In this regard it is believed that the recommendations of this panel represent viable actions that favorably impact DOD PDSS capabilities.

The panel reviewed many issues and arrived at eight specific recommendations. The eight recommendations are, in order of importance, as listed below.

Actions/Recommendations (in order of importance).

- DOD software support policy - Review and promulgate DOD-STD-1467 across the Services.

- PDSS software commonality office - Establish an office, within the Service logistics activities, to provide centralized support and coordination and the identification, procurement and distribution of software tools.

- DOD PDSS policy - Develop and promulgate a DOD Instruction on PDSS Policy.

- PDSS training for managers - Implementation of three courses for program managers: initially a short video tape followed by first a one day tutorial and eventually a two day tutorial.

- Software support environments and tools - Promulgate DOD-STD-1837 (CAIS) across the Services.

- Modernization of tools and technology for PDSS of pre-Ada systems - In addition to on-going Ada programs, pursue (establish) an other-than-Ada software engineering technology program.

- Acquisition Regulation support - Support the activity of the DAR Subcommittee on Technical Data Rights, particularly for a new Rights in Software clause.

- Ada conversion criteria - Establish a program to develop criteria for the conversion of existing software to Ada.

Although the panel believed the above to be in order of most impact, a more practical or implementable priority was developed based on other factors such as ease of implementation. The panel, therefore, developed a priority algorithm based on:

- Ease of JLC ability to direct the implementation of the recommendations.

- Impact on the PDSS.

- Ease of implementation.

The priority which resulted was:

1. Promulgate DOD software support policy.

2. Establish PDSS software commonality office.

3. Promulgate software support environment standards.

4. Improve Acquisition Regulation support.

5. Promulgate DOD PDSS policy.

6. Improve PDSS training for managers.

7. Modernize tools/technology for PDSS of pre-Ada systems.

8. Develop Ada conversion criteria.

It is expected that the first, DOD software support policy, and the third, CAIS implementation, will prove most contentious. The first, DOD software support policy, and the fourth, Acquisition Regulation support, can be accomplished quite easily.

The recommendations that require outside JLC action will be most difficult or time consuming are the fifth, sixth and seventh, Promulgating DOD PDSS policy, Improving PDSS training for managers, and Modernizing tools/technology for PDSS of pre-Ada systems.

The last of these should probably be accomplished by an expanded STARS program, in line with the original intent and objectives of that program.

While this priority may need to be modified, it was felt that the recommendations were sound and represented real actions that could be taken to assure a positive impact on the supportability of DOD systems.

THE CHALLENGE.

The implementation of a viable PDSS environment for the Services represents a continuing challenge. This challenge is even greater when primary emphasis is placed on acquisition over life cycle maintenance -- resulting in the myriad of policies and technology programs directed at software development issues in the belief that these will also provide for the post deployment phase. While this is partially true, the post deployment phase has its own unique problems, one of which is technology transition. We believe that the program of recommendations presented herein represents a sound start at inserting effective technology transition into PDSS capabilities.

SOFTWARE TECHNOLOGY TRANSITION SUBPANEL MEMBERS.

Co-chairs:	Marciniak, John	Marciniak and Associates
	Holinko, Myron	Army (CECOM-CLCSE)

Subpanel 1.

Chairman:	Murray, William (Mac)	General Dynamics
Members:	McDonald, James	Air Force
	Preston, David G.	IIT Research Institute
	Smith, Jerry	QSOFT

Subpanel 2.

Chairman:	Nuhn, Perry R.	Software Productivity Consortium
Members:	Bedar, George, Maj	Marine Corps
	Bracker, Lynne	Hughes
	Calland, Robert	Navy (NOSC)
	Glushko, Robert	SEI
	Irwin, Allen T.	SAIC

Subpanel 3.

Chairman:	Cooper, Jack	Anchor Software Management
Members:	Baker, Emanuel R.	Software Engineering Consultants, Inc.
	Bates, Wayne	Air Force
	Harvey, Lawrence	Teledyne Brown Engineering
	Malinowski, Greg	Army (CECOM-CLCSE)
	Rodriguez, Albert	Army (CECOM-CLCSE)
	Wasilausky, Robert	Navy (NOSC)

(Intentionally Blank)

PANEL VII LIST OF BRIEFINGS.

1. "Service Practices on PDSS, Army, Navy" - Jack Cooper, Anchor Software Management, Ltd, (703) 578-3200.

2. "DOD-STD-2167(A) Proposals" - Capt Richard Schmidt, AFSC/PLRP, (301) 981-5731.

3. "NAVAIRSYSCOM TECR Policy" - Tom Smith, NAVAIRSYSCOM, (202) 692-7035.

4. "Ada in Mission Critical Computer Systems" - Wolfhart Goethert, IIT Research Institute (IITRI), (315) 336-2359.

5. "DOD-STD-1467" - Chuck Gordon, CACI, (703) 276-2838.

6. "PDSS Data Rights" - Anne Martin, Software Engineering Institute (SEI), (412) 268-7622.

7. "Software Engineering Environments" - Hank Stuebing, Naval Air Development Center, (215) 441-2314.

8. "Service Practices on PDSS, Air Force" - John Marciniak, Marciniak and Associates (informal discussion, no charts presented), (703) 920-9116.

(Intentionally Blank)

PANEL VII BIBLIOGRAPHY.

1. "Adequate Planning for Acquiring Sufficient Documentation About and Rights in Software to Permit Organic or Competitive Maintenance", SEI-86-TM1, Software Engineering Institute, Samuelson, P., March 1986.

2. "Army PDSS Charter - Life Cycle Software Implementation Plan" HQ's DA, December 1983.

3. "Automated Environments and Software Quality Assurance: The Coming Crisis", Baker, Emanuel R., Software Engineering Consultants, Inc.

4. "DOD-HDBK-287 (Draft), Defense System Software Development Handbook", 23 May 1986.

5. "DOD-STD-1467 (AR), Software Support Environment", January 1985.

6. "Final Report of the JLC Workshop on PDSS for MCCR, Volume I, Executive Summary", June 1984.

7. "Final Report of the JLC Workshop on PDSS for MCCR, Volume II, Workshop Proceedings", June 1984.

8. "IDA Report on Technology Transition".

9. "Proposal for a New "Rights in Software" Clause for Software Acquisitions by the DOD", Technical Report CMU/SEI-86-TR-2, Samuelson, et al., Software Engineering Institute, September 1986.

10. "Report on the Life Cycle Impact of Ada on US Army Systems".

11. "Software Technology Transition & Support Plan", AFSC, December 1985.

12. "Toward a Reform of the Defense Department Software Acquisition Policy", Technical Report CMU/SEI-86-TR-1, Software Engineering Institute, Samuelson, P., April 1986.

13. "Understanding the Implications of Selling Rights in Software to the Defense Department: A Journey Through the Regulatory Maze", SEI-86-TM3, Software Engineering Institute, Samuelson, P., March 1986.

(Intentionally Blank)

MCCR SECURITY
PANEL VIII
PROCEEDINGS

PURPOSE.

Security accreditation is a costly and labor intensive effort. Current directives are incomplete, inconsistent, and do not adequately consider the impact of security requirements implementation.

OBJECTIVE 1.

The MCCR Security in PDSS Panel examined how strong technical security requirements will affect PDSS activities, and how strategies for incorporating security requirements can be incorporated into PDSS planning. This Panel will produce specific recommendations for the incorporation of security requirements into existing Agency and Military Department computer security guidelines and directives.

APPROACH.

The Panel approached the problem by defining and addressing five major topics:

1. Subtask A - Establish a list of security requirements necessary to satisfy MCCR security and safety objectives.

2. Subtask B - Map MCCR security requirements to existing industry and Service computer security regulations and guidelines.

3. Subtask C - Identify deficiencies with current industry and Service computer security regulations and guidelines.

4. Subtask D - Identify where future research and development should be focused.

5. Subtask E - Identify a standard set of software metrics that provide measurements for a technical assessment of the extent to which security requirements are met.

FINDINGS/DISCUSSION.

General. The single largest improvement in MCCR security can be achieved by integrating security requirements into the system engineering discipline. However, additional tools are required to satisfy and maintain the higher levels of trust necessary for more critical applications. The Panel emphasized in the strongest possible terms that major improvements in system

security can be made by including security requirements at the beginning of a project. This may allow the use of existing software tools to satisfy both security requirements in addition to "conventional" software development requirements. Satisfying security requirements is a requirements definition process that demands a rigorous application of sound systems engineering disciplines including a comprehensive quality assurance program.

Subtask A.

The identification of computer security requirements is dependent on the system application. For information processing systems, a secure system is one that guarantees the integrity of and proper access to the information. For process oriented systems, such as a weapons control system, security means ensuring that the weapon is trustworthy and will perform as intended; that it is aimed correctly, that it goes where it is supposed to, that it is not inadvertently fired, and that it is resistant to in-transit countermeasures. For control systems, such as a navigation system, the security aspect may be that the system always works (reliability).

Issues. Four issues were raised with respect to a PDSS Center:

1. Certification and accreditation of a system over its life cycle.

2. Accreditation for an existing (nonaccredited) system.

3. The impact of hardware maintenance for a secure system.

4. The impact of secure software maintenance and distribution.

Definitions. Definitions of certification and accreditation are as follows:

1. Certification is the process of ensuring that an operational system precisely satisfies specified (security) criteria.

2. Accreditation is a determination by proper authority, the Designated Approving Authority (DAA) that the operational system works well enough so that the operational need for the system outweighs the operational risk associated with system deployment when evaluated against the certification criteria.

The application of these definitions revealed three states for a system:

1. Deployed; not certified or accredited.

2. In development; security requirements not identified.

3. New starts; certified and accredited.

Activities. MCCR security activities for PDSS should be chosen in a manner that is independent of, but takes into consideration, applicable certification criteria. The set of activities to be applied to PDSS is determined by the state of the MCCS. Variants of the set are the state of certification and accreditation of the system and the PDSS activities necessary to satisfy MCCR security requirements. The list of activities include:

1. Security certification and accreditation determination (all states).

2. Security enhancement assessment plan (deployed and in-process states only).

3. Security accreditation plan (all states).

4. Security certification package (all states).

5. Independent verification and validation documentation.

Subtasks B and C.

Current regulations, guidelines and policy directives associated with security and mission critical systems deal primarily with information processing security and do not adequately address process security. The principal current regulation regarding computer security is DOD-STD-5200.28 ("Orange Book"). The Orange Book provides certification criteria and requirements for general purpose operating systems that must support DOD information security policy; it requires interpretation for application and it does not provide a complete baseline for certifying or accrediting process control systems that require a different interpretation of the term "security".

Issue. The Orange Book is a standard. Guidance for its use and application is inadequate. The Orange Book does provide a baseline for the derivation of criteria for other information processing security applications and for process control security applications. This baseline is provided by the theory and rationale contained in it, but other interpretations must be developed for data base, network, and process control application areas.

The support environment that exists in a PDSS center is usually a general purpose computer system whose resources are applied to the problem of providing life cycle support for a mission critical system. Such a system falls into the category of an information processing system for which the Orange Book criteria

are applicable. However, further guidance is needed and the Naval Research Laboratory (NRL) Report entitled "An Approach to Determining Computer Security Requirements for Navy Systems", by Carl Landwehr and H. O. Lubbes, is an example of this guidance.

Subtask D.

Research and development efforts from which near term, mid term, and long term benefits could accrue were identified.

Near Term. In the near term (applicable to deployed and in-development systems), the following efforts are suggested:

1. The development of security specific testing tools and methods that include penetration packages, regression test support, stress testing, and code analysis tools.

2. The definition and development of a standard evaluation process.

3. The adaptation and application of existing software engineering tools.

Mid Term. Those projects for which mid term (3 to 5 years) benefits can be expected included:

1. Security modeling for the solution space, the threat, and the necessary analysis.

2. The application of knowledge base technology to the automation of the software development process supporting the transition between development life cycle phases while preserving the complete traceability and providing (semi-) formal verification for the system.

Long Term. Long term benefits can be expected from hardware and architecture efforts. The Panel noted that mid and long term benefits would be realized for PDSS of new starts.

Subtask E.

Metrics that could be applied to the determination of the extent to which security requirements are met included:

1. The extent to which a (disciplined) development approach was followed.

2. The extent to which the specific security evaluation criteria are satisfied.

3. Based on the application of code analysis tools and techniques, code quality, the presence or amount of "dead" code, and the complexity of the code.

4. The extent to which the system is modularized and the degree to which security-critical code is isolated.

5. The anticipated amount of difficulty to accredit the system as a function of the perceived complexity of the system.

6. All standard software engineering quality metrics.

RECOMMENDATIONS.

Subtask A.

Recommendation 4-8-01. JLC JPCG-CRM develop and coordinate a security awareness and training program for Project Managers and PDSS operational personnel.

Subtask B and C.

1. Recommendation 4-8-02. Strict systems and software engineering standards must be defined and enforced throughout the life cycle (development and post deployment) of the system.

2. Recommendation 4-8-03. Determine the Designated Approving Authority (DAA) at the beginning and involve the DAA throughout the life cycle of a "secure" system.

3. Recommendation 4-8-04. Risk management must be a continuous process from requirements definition throughout the life cycle.

4. Recommendation 4-8-05. Provide full IV&V documentation to the PDSS Center. This documentation is vital to the post deployment support process.

5. Recommendation 4-8-06. Under the auspices of NSDD-145, the National Telecommunications and Information System and Security Committee must establish a single source for DOD computer security policy. The variety of existing DOD computer security policies and guidelines must be integrated into a single cohesive set, eliminating the confusion caused by conflicting direction.

6. Recommendation 4-8-07. A mechanism for assessing the impact of a change to a system on the security of that system must be defined. A necessary part is the placement of the DAA on the configuration control board for the system.

7. Recommendation 4-8-08. The Orange Book requirements must be interpreted and guidance provided to address networks, data bases, and process control security applications as well as information systems security applications other than operating systems.

8. Recommendation 4-8-09. The JLC should establish a committee to develop changes to DOD-STD-2167 that incorporate security requirements as an integral part of a systems development life cycle. The standard must include specific Service requirements as well as NCSC requirements; it must provide DIDs to detail the required deliverables; and it should be augmented by a guidebook for application of the security standards. (A starting point for such a guideline is the "Computer Security Acquisition Management Guidebook," developed by the Space and Naval Warfare Systems Command.) The basis for this standard should proceed from an appropriate modification to DODD's 5000.1 and 5000.2.

9. Recommendation 4-8-10. The Services must have an organic capability to evaluate systems against trusted computing criteria and certify them for the accreditation process. (The certification process provided by the NCSC takes too long.)

10. Recommendation 4-8-11. The JLC should take steps to expedite the development and release of network certification criteria and data base evaluation criteria.

11. Recommendation 4-8-12. The JLC should expedite the completion and release of standard language regarding security requirements for inclusion in contracts and SOW.

12. Recommendation 4-8-13. Establish specific guidelines that address the security requirements for the transition of a system to a PDSS Center.

Subtask D.

1. Recommendation 4-8-14. The government must provide support for verification tools that are to be used for trusted systems development.

2. Recommendation 4-8-15. Establish a Security Efforts Coordination Agent under the JLC JPCG-CRM to make maximum use of individual Service security efforts.

Subtask E.

Recommendations derived as a result of this subtask were forwarded to Panel V, PDSS Management Indicators and Quality Metrics.

IMPACTS.

Continued use of systems which cannot be trusted to maintain separation of data of different classification or sensitivity, or protect processes from the action of critical or untrusted processes, or which cannot protect critical processes or sensitive data from unauthorized access or tampering by human operators will prevent us from realizing the full potential of available computing capability. We will continue to have to use separate computer systems to process information of different sensitivity levels with the attendant costs of separate, duplicate hardware and the restriction that human interfaces between systems of unequal sensitivity levels have on data sharing among systems. We will continue to have an unacceptable level of confidence when computer systems which control weapons or safety critical devices cannot be disabled or caused to operate other than intended because of the vulnerability of critical processes to other processors, bad data, or out of sequence commands.

BENEFITS.

Embedding computer security requirements into DOD-STD-2167 will establish computer security as a development discipline across DOD. Guidance on defining application specific computer security requirements and carrying out computer security functions during the life cycle will support the requirements in DOD-STD-2167. Service organic capabilities for evaluating products against the trusted computer base (TCB) criteria and certification of the applications systems ability to satisfy selected security requirements will speed up the evaluation and certification process. The R&D recommendations are focused on providing improved tools for satisfying requirements and providing greater assurance that we can trust our security implementations.

A longer range recommendation focused on providing systems architecture to better support security for real-time process control functions. All of the recommendations are aimed primarily at the system acquisition process because it has been shown that attempting to retrofit security is both costly and largely ineffective.

The greatest benefit to PDSS activities is for those activities to begin with an accredited system and the tools used to certify that system's compliance with security requirements. These tools are necessary to maintain the system's compliance with its security requirements.

OBJECTIVE 2.

To identify deficiencies with the DOD Security Program and recommend modifications to security regulations and industrial guidelines. Focus will be on the effect that strong technical security requirements have on PDSS activities as well as strategies for incorporating security requirements into the PDSS planning phase.

Background. The subgroup was divided into five panels whereby each panel was designated a particular subject area to investigate. Panel VIII's objectives and goals were to identify deficiencies with the DOD Security Program, recommend modifications to security regulations and industrial guidelines, identify multilevel security requirements for MCCR, identify future R&D efforts, map MCCR security requirements to their specific objectives and to identify software metrics that measure the extent to which security requirements are being met.

ASSUMPTIONS AND CONSTRAINTS.

1. Security accreditation is costly and labor intensive.

2. Current directives are incomplete, inconsistent and do not adequately consider operational impacts with security requirements implementation.

3. The identification of computer security requirements is dependent on the system application. For example:

a. For information processing systems, a secure system is one that guarantees the integrity of and proper access to the information.

b. For process oriented systems, such as a weapons control system, security means ensuring that the weapon is trustworthy and will perform as intended, e.g. it is not inadvertently fired, it goes where it is supposed to, it is aimed correctly, and it is resistant to in-transit countermeasures.

c. For control systems, such as a navigation system, the security aspect may be reliability, that the system always works.

d. Four issues were raised with respect to a PDSS Center:

(1) Certification and accreditation of a system over its life cycle.

(2) Accreditation for an existing, nonaccredited system.

(3) Hardware maintenance impact for a secure system.

(4) Secure software maintenance and distribution impact.

e. The definition of certification and accreditation are:

(1) Certification: The process of ensuring that an operational system precisely satisfies specified (security) criteria.

(2) Accreditation: A determination by proper authority, the Designated Approving Authority (DAA), that the operational system works well enough, so that the operational need for the system outweighs the operational risk associated with system deployment, when evaluated against the certification criteria.

f. The application of these definitions revealed three states for a system:

(1) Deployed; not certified or accredited.

(2) In development; security requirements not identified.

(3) New starts; certified and accredited.

g. MCCR security activities for PDSS should be chosen in a manner that is independent of but takes into consideration applicable certification criteria. The set of activities to be applied to PDSS is determined by the state of the MCCS. The variants of the set are the state of certification and accreditation of the system and the PDSS activities necessary to satisfy MCCR security requirements. The list of activities include:

(1) Security certification and accreditation determination (all states).

(2) Security enhancement assessment plan (deployed and in-process states only).

(3) Security accreditation plan (all states).

(4) Security certification package (all states).

(5) IV&V documentation.

4. Current regulations, guidelines and policy directives associated with security and mission critical systems deal primarily with information processing security and do not adequately address process security.

SCOPE.

Panel VIII's efforts were directed at integrating security requirements into the system engineering discipline. The consensus of panel members was that the MCCR security challenge for PDSS could most significantly be met through the incorporation of computer security requirements into the life cycle of a MCCR system. The chart entitled "Acquisition Life Cycle Technical Activities" shows clearly where and how security should be incorporated into the system life cycle process.

APPROACH.

The panel approached its objective by first discussing and examining various relevant topics that would provide insight into the MCCR security environment. Issue papers were written. Following examination of several topic areas, the panel agreed on the major issues and devised their specific recommendations to propose to the JLC. These issues and recommendations are discussed in the body of this report.

Product 1: Recommended Modifications to Current Industry and Service Regulations and Guidelines.

DISCUSSION.

Security accreditation is a costly and labor intensive effort. Current directives are incomplete, inconsistent and do not adequately consider operational impacts with security requirements implementation. The Services are currently duplicating their efforts by developing individual computer security guidelines which is both costly and confusing to defense contractors who support more than one Service. Refer to the detailed product section below for the panel's specific recommendations.

COGENT FACTORS.

Return on Investment. The monetary costs to modify and/or establish regulations and guidance is less than the costs that would be evident should the security of classified data be compromised.

Dependencies upon other Actions/Recommendations. Prior to modification of current guidance, JLC must ensure that under the auspices of NSDD-145 the National Telecommunications and Information System and Security Committee establish a single source for DOD computer security policy.

List of Alternatives. None.

Method of Implementation. The JLC should establish a committee to develop changes to DOD-STD-2167 that incorporate security requirements as an integral part of a systems development life cycle. The standard must include specific Service requirements as well as NCSC requirements; it must provide DIDs to detail the required deliverables; and it should be augmented by a guidebook for application of the security standards. (A starting point for such a guideline is the "Computer Security Acquisition Management Guidebook", developed by the Space and Naval Warfare Systems Command.) The basis for this standard should proceed from an appropriate modification to DODD's 5000.1 and 5000.2.

Justification. No further justification is needed other than to state that there is a lack of guidance, that there are MCCR systems processing classified data and further development is currently occurring.

Detailed Product. For the purpose of clarification, the following recommendations have been grouped accordingly.

1. EMBED COMPUTER SECURITY REQUIREMENTS IN DOD-STD-2167.

a. Strict systems and software engineering standards must be defined and enforced throughout the life cycle (development and post deployment) of the system.

b. Determine the DAA at the beginning and involve the DAA throughout the life cycle of a "secure" system. Refer to "Role of the DAA In the Acquisition and Operation of Systems" for a thorough discussion of DAA roles and responsibilities.

c. Risk management must be a continuous process from requirements definition throughout the life cycle.

2. DEVELOP A COMPUTER SECURITY IMPLEMENTATION GUIDEBOOK.

3. DEVELOP BETTER GUIDANCE ON IDENTIFYING SECURITY REQUIREMENTS.

a. The Orange Book requirements must be interpreted and guidance provided to address networks, databases, and process control security applications as well as information systems security applications other than operating systems.

b. A mechanism for assessing the impact of a change to a system on the security of that system must be defined. A necessary part is the placement of the DAA on the configuration control board for the system.

4. OTHER.

a. The JLC should take steps to expedite the development and release of network certification criteria and data base evaluation criteria.

b. The JLC should expedite the completion and release of standard language regarding security requirements for inclusion in contracts and SOW.

c. Establish specific guidelines that address the security requirements for the transition of a system to a PDSS Center. Refer to "PDSS Environment" for specific recommendations regarding transitioning a MCCR system to a PDSS center.

Product 2: List of Recommended Multilevel Security Requirements.

DISCUSSION.

The list of recommended multilevel security requirements as shown below is extracted from DOD-STD-5200.28. The panel agrees that this list of requirements meets the MCCR security requirements and safety objectives. Action is required to establish guidance on the methodology for determining the requirements for a MCCR system. NRL Report No. 8897 provides such guidance. Refer to Issue Paper "DOD-STD-5200.28 (Orange Book) Analysis" for a thorough discussion of the problem.

List of Multilevel Security Requirements.

Audit
Configuration Management
Covert Channel Analysis
Design Documentation
Design Specification and Verification
Device Labels
Discretionary Access Control
Exportation of Labeled Information
Exportation to Multilevel Devices
Exportation to Single Level Devices
Identification and Authentication
Label Integrity
Labeling Human Readable Output
Labels
Mandatory Access Control
Object Reuse
Security Features Users' Guide
Security Testing
Subject Sensitivity Labels
System Architecture
System Integrity
Test Documentation
Trusted Distribution
Trusted Facility Management
Trusted Facility Manual
Trusted Path
Trusted Recovery

Product 3: Comparison of Security Requirements to the Orange Book, OPNAV 5239.1A and Other Service and Industry Guidelines.

DISCUSSION.

The panel agreed that the security requirements identified in DOD-STD 5200.28 (Orange Book) satisfied MCCR system security objectives. Security documentation and guidance including the Army document AR 380-380, Navy Document OPNAVINST 5239.1A, Air Force Document AFR 205-16 and the World Wide Military Command and Control System documents were compared to the listed security requirements to identify if the guidance in the documentation adequately supported the security requirements. It is clearly demonstrated that current Service guidelines are inadequate in providing DOD with consistent instruction.

Product 4: Prioritized List of Areas that Require Further Research and Development.

DISCUSSION.

Computer Security is recognized as a viable R&D requirement, although program policy and direction including adequate funding have not been forthcoming. The JLC must ensure that computer security R&D for MCCR be given the appropriate attention. Refer to "MCCR Security R&D" for a complete discussion of MCCR R&D issues. The current process of certification provided by the National Computer Security Center is extremely long, cumbersome and inadequately staffed. Further, should system design incorporate trusted featured PDSS facilities have neither documentation or automated tools with which to maintain an adequate security posture during a change process. The detailed products section below provides specific recommendations concerning the future focus of MCCR R&D. "Establish a MCCR Security Baseline for PDSS" addresses necessary activities over and above existing R&D programs. This paper assumes the continuation of existing and planned Computer Security R&D efforts of the National Computer Security Center, the Services and DOD agencies.

COGENT FACTORS.

Return on Investment. The monetary costs to adapt existing software tools and/or to develop new tools which are necessary for verification & validation of security requirements in MCCR systems is less than the costs should t security of classified data be compromised.

Dependencies on other Actions/Recommendations. To centralize R&D efforts, the JLC must ensure a single source for DOD Computer Security policy is established under the auspices of NSDD-145, the National Telecommunications and Information System and Security Committee.

List of Alternatives. None.

Method of Implementation. Further analysis shall be required by the appointed single point of contact to determine the method of implementation to provide the program, in as short a period of time as possible, efficient software tools to use.

Detailed Products. The panel recommends that an adequate MCCR R&D security program be established and sufficiently funded to provide technical solutions specific to meet MCCR needs. R&D efforts from which near term, mid term, and long term benefits could accrue are identified below.

1. Near Term. In the near term (applicable to deployed and in development systems), the following efforts are suggested:

a. The development of security specific testing tools and methods that include penetration packages, regression test support, stress testing, and code analysis tools.

b. The definition and development of a standard evaluation process.

c. The adaptation and application of existing software engineering tools.

2. Mid Term. Those projects for which mid term (3 to 5 years) can be expected include:

a. Security modeling for the solution space, the threat, and the necessary analysis.

b. The application of knowledge base technology to the automation of the software development process supporting the transition between development life cycle phases while preserving the complete traceability and providing (semi-) formal verification for the system.

3. Long Term. Long term benefits can be expected from hardware and architecture efforts. The Panel noted that mid and long term benefits would be realized for PDSS of new starts.

4. The government must provide support for verification tools that are to be used for trusted systems development.

5. Establish a Security Efforts Coordination Agent under the JLC JPCG-CRM to make maximum use of individual Service security efforts.

6. Provide full IV&V documentation to the PDSS Center. This documentation is vital to the post deployment support process.

7. The Services must have an organic capability to evaluate products and applications against trusted computing criteria and certify them for the accreditation process. The certification process provided by the NCSC takes too long.

Product 5: Identify Standard Set of Software Metrics That Provide Measurements for a Technical Assessment of the Extent to Which Security Requirements are Met.

DISCUSSION.

Recommendations derived as a result of this subtask were forwarded to Panel V, PDSS Management Indicators and Quality Metrics. "Security Evaluation of Existing Systems In The PDSS Environment" discusses issues that relate to software metrics.

PANEL VIII BRIEFINGS - PART 1
PDSS ENVIRONMENT

1. The software security features of the existing systems delivered to the PDSS for life cycle support must have, as a minimum, the security level of the originally CERTIFIED SYSTEM maintained at the same level. The Designated Approving Authority (DAA) should insure that the same tools utilized in the development of the secure system are delivered to the maintaining facility with the appropriate training and documentation. These would include the same facilities used in the development of software not security related (e.g., compilers, editors, etc.). In addition, this would include the formal security model (if applicable), Detailed Top Level Specifications, Formal Top Level Specifications, Rationale Documentation, Security Verification Environment special hardware that supports the verification process, secure configuration management plan and the capability to keep current the analysis and documentation required to stay certified or obtain recertification so that the DAA can continue to accredit the system.

2. The unique tools that are security related such as formal verification languages, theorem provers or secure configuration management tools would have to be maintained and used in the PDSS Center to insure the integrity of the security features are preserved.

3. If a change is proposed to the trusted computer base, the original certification criteria would have to be invoked and the same steps and procedures would have to be exercised on the proposed changes.

4. In addition to the security tools and verification environment, if special constraints were placed on the original developer such as special personnel clearances or classification of code or specifications, the same constraints would have to be enforced on the PDSS environment.

(Intentionally Blank)

PANEL VIII BRIEFINGS - PART 2
DOD-STD 5200.28 (ORANGE BOOK) ANALYSIS

1. The Orange Book is the result of ideas originated by the DOD Computer Security Institute. These ideas centered around the need to have industry provide the computer hardware and system software (operating systems (OS) and utilities) in a competitive environment, which the DOD needs to support secure applications. In order to get industry to provide these TCBs, DOD had to define its requirements and provide criteria for evaluating industry response to the requirements. The Orange Book defined the requirements and provides the criteria. Specifically, the requirements were for general purpose computers which would enforce varying degrees of DOD security policy with varying degrees of assurance when used in an environment characterized by the typical general purpose time sharing system of the late 70's and early 80's. That is, a system consisting of a host processor with directly connected remote unintelligent terminals providing among other Services on-line programming.

2. The question is: Does the Orange Book apply to application systems and if so how? The Orange Book provides the only applicable consistent, hierarchical expression of the requirements for a TCB. It is the best available tool for defining the hardware and software requirements for secure application systems but provides no internal guidance regarding which requirements apply to a given application. The totality of the security requirements of an application must be met by the combination of the system software and the application software. In an ideal use (one in which TCBs meeting a wide range of TCB criteria are available) the bulk of the trusted functions required by the application are met by the available general purpose TCB and the amount of application specific trusted software is small. When a secure application must be implemented on an untrusted computing base or one which does not meet the bulk of the requirements, most of the trusted software must be provided by the application. Because of the expense of developing application unique trusted software, care must be taken to define the minimum acceptable requirements. For an application system which meets the previous definitions of a general purpose time-sharing system, requirements may be identified from the Orange book by simply quantifying the difference between the highest classification of data processed by the system and the clearance of the lowest cleared user of the system as proposed by the draft DOD 5200.28. When the application differs from the general purpose time sharing environment, additional factors which express operational risk must be evaluated to select a set of requirements from the Orange Book.

3. An initial step in defining the additional factors which express operational risk has been taken. This step has been documented in NRL Report No. 8897. The NRL report identifies three risk factors in addition to the difference between classification of data and clearance of the user. These factors are: local processing capability, communication path (user to host), and user capability (provided by the host). Use of these factors provide a means of a variance from the environment upon which the Orange Book was based.

4. Additional work must be done to refine the process of selecting requirements from the Orange Book. This work has been funded and is being placed on a delivery order contract with Logicon, Inc. Refining the process of mapping specific application to the Orange Book will proceed in three dimensions. The first is to identify additional risk factors and the relative weights which should be assigned those factors. The second is to identify the contribution each of the criteria within the seven levels of trust in the Orange Book make to the level of trust. This effort will also identify which criteria are independent and which must be used in conjunction with other criteria to accomplish some level of trust. This dimension of the effort is to identify additional meaningful levels of the criteria because it is difficult to map all applications to only seven levels of trust. The third dimension of the work is to identify trade-offs that can be made by using communication, procedural, and physical security measures in place of hardware and software measures. The goal of this effort is to provide a more comprehensive tool for mapping application specific security requirements to the Orange Book.

5. It needs to be noted that in some respects the need for this refinement, especially that part of the effort to identify additional meaningful levels within the Orange Book may be temporary. As the availability of efficient TCB products increases to the point that a wide range of these products can be acquired competitively, it will then only be necessary to identify the minimum acceptable level of trust required by the application from the existing levels within the Orange Book and procure a TCB which exceeds that level. Until then, however, significant portions of required trusted software must be application specific and the cost must be assumed by the application developer.

PANEL VIII BRIEFINGS - PART 3
ROLE OF THE DAA IN THE ACQUISITION AND OPERATION OF SYSTEMS

1. According to the draft DODD 5200.28 and the Navy Computer Security Acquisition Management Guidebook, the following roles and responsibilities are proposed for the DAA throughout the system life cycle:

a. During the acquisition and development phase, DAA will:

(1) Review user security requirements to ensure that they comply with established policy and guidelines.

(2) Review and approve security specifications and requirements in RFPs.

(3) Evaluate and approve the Computer Security Accreditation plan through coordination with the Certification Authority.

(4) Review and approve operational procedures for the systems.

(5) Review and concur with the security implementation decisions made by the Project or Program Manager at each stage of development.

(6) Identify security deficiencies and, as necessary, require allocation of resources to correct those deficiencies.

(7) Review and concur with the Certification Test and Evaluation Plan.

(8) Approve with Certification Authority participation, the configuration plans, risk analysis evaluation, vulnerability assessment report, security architecture, test and evaluation report, and contingency plans.

(9) Approve formal model of the security policy and formal top level specifications of the system.

(10) Validate security requirements at the beginning of the Demonstration/Validation Phase.

(11) Approve risk evaluation after the Systems Design Review.

(12) Participate in all design reviews to ensure that security is being adequately addressed.

(13) Approve PM's evaluation of performance, cost, and risk prior to full-scale development phase.

(14) Issue interim approvals for testing of system in the Demonstration/Validation Process.

(15) Establish data ownership accountability or process responsibility to include access rights and special handling and coordinate all decisions concerning risks to the process or data with the owner(s).

b. During the operational phase, the DAA will:

(1) Accredit the system using the plans and reports derived during the development phases.

(2) Issue special restrictions to offset recognized deficiencies and direct changes as needed to correct those deficiencies.

(3) Ensure that the system continues to be operated and maintained according to the accreditation guidelines.

(4) Participate in the Configuration Control Board or assurance groups to ensure continued compliance with security through the change process.

(5) Evaluate and approve changes to systems, procedures and environment after security impact has been assessed.

(6) Ensure that a continual program of security training and awareness is in place.

(7) Approve changes to security documentation for the system.

(8) Ensure that system security implementation is reassessed at least every 3 years.

THE DAA ACCREDITATION PROCESS FOR EXISTING SYSTEMS

1. There is a subtle difference between the DAA involvement and process for the accreditation of existing and new systems. The DAA involvement in the development of new systems works toward acquisition of systems that can be "trusted" to fully protect the data, system resources, or critical processes entrusted to the system consistent with the operational environment and need. The DAA process for existing systems is of necessity more subjective and oriented toward achieving and maintaining a "reasonable level or risk" until appropriate security enhancements can be made or until the system is replaced with a new trusted capability. The "reasonableness of risk" approach implies that systems with less than fully acceptable levels of security may be accredited to operate simply because there are no alternatives that will provide for both mission accomplishment and full security. This "real life" approach is necessary in some cases because the system is already installed and operating to meet essential mission requirements and that termination of those services would have a disastrous effect upon that mission. In other words, the cure may be worse than the disease itself. The approach is also driven by the cost and time of retrofitting security into existing systems, our inability to verify the adequacy of the existing hardware/software security implementation in nonrated systems, and our inability to meaningfully correct security deficiencies for lack of tools and documentation or due to the complexity of the software. The "reasonable level of risk" process involves systems evaluation and analyses, review of historical data, comparison with similar systems, and adjustment of supporting security controls (mode of operation, physical security, personnel access measures), or the addition of new software until the desired level of confidence is obtained. This process recognizes that perfect security cannot be achieved considering the many unknowns of the existing configuration. It also recognizes that any work expended toward the correction of areas of deficiency or "distrust" must be consistent with the projected system life cycle.

2. Depending upon the staffing and concept of operations of a particular DAA activity, either the DAA or the operational activity may be the driver of the process to secure DAA accreditation of the existing system. The process must begin with a "snapshot" of the system and security implementation as it stands at a particular point in time. This snapshot will be documented as a security plan or computer security accreditation plan to describe the system hardware/software configuration, intended purpose of the system, mode of operation, environment, existing support controls (physical, personnel, TEMPEST, Communications Security, system or file access or process control, audit capability), contingency and emergency plans, and procedures. The assistance of functional experts at the operational site is necessary to fully document the needed plan.

3. Since there will be no existing certification for the hardware/software configuration, the operational activity must attempt to establish the "trustworthiness" or "level of confidence" for any nonrated system. This will involve the accumulation and analysis of a wide range of documents and historical data such as: tools or techniques used in system development, IV&V results, controls exercised during software development process, languages and compilers used, controls over system changes made after deployment, hardware/software problems encountered since deployment at this or similar sites, known deficiencies in basic operating systems or utility software, user group experiences, etc. Code analyses or evaluation by the agency evaluation authority would add a degree of confidence but the costs of those additional processes must be weighed against benefits considering the expected system life cycle before they are undertaken. They must also be based upon a demonstrated need after the trade-off process clearly shows that the alteration of other security controls cannot offset identified deficiencies and still allow for effective accomplishment of the mission. The DAA representative should at least assist in the quasi-certification of the system by providing experience from similar efforts using similar systems. The operational activity becomes the "de facto" certification authority to the DAA.

4. A risk analysis must then be performed by the activity, possibly with DAA assistance, to identify threats, system vulnerabilities, pair those threats and vulnerabilities to identify risks, assess or devise countermeasures to offset the risks, and identify residual risks.

5. A security test and evaluation plan must then be prepared and exercised for actual testing of security countermeasures or controls to demonstrate the adequacy of the implementation. Testing of critical controls and countermeasures, however rudimentary, must be accomplished to ensure that those controls or countermeasures accomplish the intended results. The risk analysis results may be affected and require reworking where established countermeasures prove to be ineffective.

6. Finally, all of the information identified in the above processes (security plan, quasi-certification, risk analysis, and security test/evaluation report) must be provided to the DAA activity for a subjective evaluation of the "reasonableness of risk" for the overall system. The resultant accreditation may involve an interim acceptance of risk with directed corrective actions, or acceptance of the system "as is."

7. Needless to say, system configuration control and management must be applied to the system including procedures, environment, facilities, supporting security measures, hardware and software coincident with the snapshot process. From that point, all proposed changes to the system must be assessed for impact upon the security of the system. Concurrence of the DAA must be obtained before such changes can be implemented into the operational system.

(Intentionally Blank)

PANEL VIII BRIEFINGS - PART 4
MCCR SECURITY R&D

Computer Security R&D is not a new subject or endeavor. The consolidated Computer Security Program at the National Computer Security Center (NCSC) establishes a small joint Service R&D program directed at the security issues of information systems. In addition, each of the Services have minor activities addressing security needs in specific application developments. Much of this work can be applied to the MCCR computer security problems. Unfortunately, a good deal of the applicable results are either inadequate for MCCR or incomplete. There are also special aspects of MCCR security that are not being addressed by the current efforts.

Recommendation.

a. Ensure that an adequate MCCR R&D Security program is established and sufficiently funded to provide technical solutions specific to MCCR needs. The ensuing paragraphs describe a baseline program of activities believed to address the immediate needs and that would provide the biggest payoff. The program will be described in three phases: (1) near term payoff realized in 1-2 years; (2) mid term pay-off in 3-5 years; and (3) long term payoff in 6 or more years. The time of payoff assumes all activities are started at once.

b. Probably the single largest contribution to "good" computer security designs and implementations is stringent adherence to "good engineering practices". The same tools, techniques and methodologies that are used to develop quality code and systems, provides the framework necessary for trusted code. In addition, when security requirements are treated just like any other performance/functional requirement, the rigorous application of classical system engineering tools can result in the kind of internal architecture necessary to satisfy the requirements of a security Kernel. The resultant system can then be subjected to a more complete analysis and hence a higher degree of trust.

c. Systems in the existing inventory and those entering that inventory were all developed with some degree of adherence to those "good engineering practices". Therefore, the existing "tools of the trade" provide the most immediate payoff to the MCCR security problem. The proposed R&D activity includes the development and application of methodologies for adapting existing systems engineering and systems analysis tools to support the assessment of the security attributes of those existing systems. The same tools would realize an even greater payoff to systems under development.

d. The existing system engineering and system analysis tools do not take into account some of the assurance requirements necessary to satisfy the higher levels of trust identified in DOD-STD-5200.28. In some cases, satisfying specific requirements of the standard is relatively straight forward. In others, a more precise statement of the requirement is necessary in order to uniformly determine compliance with the requirements. Effort have begun to do just that. These efforts must be accelerated. Because each interpretation of the standard is unique, the certification process is extremely long and cumbersome. Currently, the National Security Agency (NSA) is providing the principal computer security certification support to the DOD. The Services are beginning to establish similar organic capabilities. However, the process as it is currently accomplished, is too time consuming and, labor intensive to simply pour more warm bodies at the problem. The process itself must be streamlined. Any efforts to streamline the process must start with a precise definition of the requirements. In fact, in order for the Services to establish organic certification capabilities, this understanding must be in place. Otherwise, one cannot hope for consistent evaluation results across the different Services and agencies. This effort is likely to identify a need for generic tool development to support analysis of security attributes not covered by adapting existing tools and techniques. Accomplishing the activities described above will result in a substantial improvement in MCCR Security and will satisfy the immediate pressing needs of the PDSS activities. With the development of a consistent, specific interpretation of the DOD-STD-5200.28, and methods for assessing compliance with those standards, we stand a chance of satisfying the growing demand for secure systems evaluations. If this process can be sufficiently well defined, it could be taken to the TEMPEST program.

e. For instance, selected contractors could be certified to perform security certifications on behalf of the DOD. These certifications would carry the same weight as if performed by DOD. This approach would dramatically increase the resources available to certify the security of computer systems and perhaps is the only way to adequately meet the demand in the future. The near term activities define a framework from which this process can be derived. The mid term activities will result in tools that better support such a process.

f. The proposed activities for payoff in the next 3 to 5 years involve sophisticated, complex development efforts requiring integration of different techniques and methodologies into cohesive security modeling and automated software engineering environments. The goal is to create tool sets that can continue to support systems through concept formulation, design, implementation, and maintenance. The same tools that are used in the development process can then be transitioned with the

system to continue the life cycle development activities usually referred to as maintenance. While these tools will not provide an immediate return to current systems entering the PDSS environment, they will contribute to the smooth and effective transition of systems in the future.

g. The efforts for the long term are technology-based issues crucial to efficient, sufficient MCCR security implementations. While the real payoffs will not be realized for six or more years, the early results will contribute to the clear definition of MCCR security requirements. Long term results will include hardware and software architectures that optimize the specific security needs of the MCCR community, and the adaptation and application of formal verification techniques and artificial intelligence to security analysis, design, and operation.

h. The proposed baseline security program addresses necessary activities over and above existing R&D programs. It assumes the continuation of existing and planned computer security R&D efforts of the NCSC, the Services and DOD agencies. Those efforts contribute to the activities described above.

(Intentionally Blank)

PANEL VIII BRIEFINGS - PART 5
SECURITY EVALUATION OF EXISTING SYSTEMS IN THE PDSS ENVIRONMENT

1. Increasingly, PDSS activities are challenged by the need to evaluate the compliance of existing mission critical software systems with the security requirements of the systems. This challenge can be met with the application of current computer security guidance and the use of existing methods and tools. The best available metric for MCCR security evaluation is the new DOD-STD-5200.28 (Orange Book). Preceding discussions on the need for guidance for applying the Orange Book criteria to application systems and the refinement of the work initiated by NRL Report 8897 present future endeavors to enhance the assessment of the security trustworthiness of MCCR systems. However, using present guidance, security requirements can be determined and security analysis can be performed to help evaluate security compliance of existing systems. Accreditation and reaccreditation needs in the PDSS environment for existing systems must be supported now.

2. The Orange Book provides a useful metric for security compliance evaluation for conceptual, developing, and existing systems. The approach for determining security requirements of an existing system is identical to that of a system to be developed. First, risk factors, or vulnerabilities, must be identified and may be based upon application of the guidance in NRL Report 8897, system environment considerations and application-specific requirements. Once the security environment definition is determined (reference Space and Naval Warfare System Command (SPAWAR) Security Acquisition Manager's Guidebook for more detailed guidance), the security requirements can be defined.

3. The ability of existing mission critical software to meet the established security requirements must be evaluated. Many current systems are not well documented and code analysis supported by security testing is the only feasible way to analyze system compliance against the defined requirements. The better the available documentation and the better the software engineering practices used for the system development, the easier the security compliance analysis will be in terms of cost, time, and increased assurance of the results. Code modularity and isolation of security critical functions are examples of good engineering practices for MCCR. Subject architectural considerations for Orange Book level compliance must be made.

4. Tools are needed to support the security evaluation effort which could be cost prohibitive if done entirely as a manual effort. There is no one security analysis tool that can examine existing source code and determine its compliance with defined security standards. However, there are many types of software analysis tools which can be applied to the security assurance effort. Two NRL Reports, one by Alan C. Shultz, using Software

Analysis Tools to Analyze the Security Characteristics of HOL Programs (July 28, 1986) and a follow-up report by Logicon, Inc., Computer Security Tools Evaluation Study (November 7, 1986) presented a first step in identifying the types of available software analysis tools which currently exist and can be applied to security evaluation tasks.

5. Software analysis tools can be used to locate potential security faults. Tools can and should be used in the development process as well as in the PDSS activity to detect requirement errors, design errors, and code errors which relate to security concerns. The Logicon/NRL tools report provides the following list of security-relevant code defects:

a. Computational errors.

b. Logic errors.

c. Input/output errors.

d. Data handling errors.

e. Interface errors.

f. Data definition errors.

g. Data base errors.

h. Coding standards deviations.

i. Malicious defects.

6. The way in which each defect might relate to a security flaw is illustrated. For example, within the Logic Errors category, an improper sequence of operations could result in a user being given access to a piece of information without the checking of access rights.

7. Computer Security Faults are categorized where a security fault (internal to the computer software) is defined as any condition or circumstance that results in the denial of service, unauthorized disclosure, unauthorized destruction or unauthorized modification of data. Security flaws are weaknesses (either by error or malicious placement) in a program that permit a security fault. Software security flaws can be sought by examining HOL application code to find where it might intentionally or unintentionally exploit operating system flaws. The Logicon report states, in terms of analyzing existing software and its modifications, that application, not System Software Source Code, is what must be considered in practice (while the software flaws are a function of the underlying system software). In one case

of an operating system product certified to the required Orange Book level, code analysis would be directed chiefly on pre-tested application processes. Categories of security flaws are identified and detailed in the Logicon report. Examples of each security fault as a software error are presented. For example, an incomplete or inconsistent parameter validation (flaw) is an example of a failure of a system to meet functional requirements pertaining to access validation and control (a traceability error).

8. Different types of software analysis can be applied to source code. Both static and dynamic (running code) analysis tools have been used to analyze existing code in development efforts. The NRL tools report presents a survey of tools which have been used to support software analysis methods. Tools include path analyzers, decompilers, flow chart generators, cross reference generators, path testing tools, interactive debuggers, execution monitors and interpretive computer simulators. The following paragraphs from page 21 of the Logicon report present possible Security Analysis Applications for existing tools:

a. "Traceability data bases may be used to catalog security requirements and to support design and code analyses to ensure that appropriate security limits have been implemented. (Now appropriate for conceptual and developing systems)."

b. "Traceability data bases may also be used in traceability analyses to ensure that all code functions are justified by proper system requirements. Detailed traceability analysis is crucial for detecting malicious security violations such as Trojan horses, trap doors, time bombs, etc."

c. "Path analyzers may be used to identify all code execution paths and to extract the conditions may then be traced back to functional and security requirements."

d. "Code analyzers can be used to test internal interface and parameter usage consistency."

e. "Data flow diagramming tools can be used to find all processes that may potentially access data objects, and to trace all potential destinations of data objects."

f. "Source and executable code comparison tools may be used to guarantee that executing code is the same as documented and analyzed code. Code comparison is necessary to provide adequate authentication of executing code."

9. The available tools for code analysis have each generally been applied to one development effort, and they are usually language specific. While some may be directly applicable to the security evaluation of an existing PDSS system, others may not. The best use of the available tools for security analysis remains an area of research which is described in the R&D portion of this paper. Results of this research are currently needed to assist in the PDSS security evaluation effort for existing mission critical systems.

10. Security testing is a necessary part of the evaluation of an existing system and is a requirement for accreditation or reaccreditation of a MCCR. Code analysis can detect specific errors, deficiencies, bad coding practices, poor structure, extraneous code, and poor control logic with respect to security requirements. Hence, code analysis against the defined security requirements provides insight into specific factors of code which will require concentrated security testing. Security testing must include checks to locate potential security weaknesses and identify program shortcomings that might indicate noncompliance with the defined security requirements. Evaluation of security test results provides valuable information for determination of the security posture of the existing system.

11. In summary, PDSS activities can perform requirements definition, code analysis, and security testing with available technology. If the software development program exhibited good software engineering practices and good documentation is available, security analysis efforts will be dramatically reduced. The PDSS activity must determine if the defined security requirements are satisfied. Code analysis tools can be used to assist in the effort to determine the degree of code complexity, and the existence of extraneous code and potential security flaws. Thorough security testing can demonstrate the security behavior of the code. Some measure of the cost/schedule impacts to the accreditation process can be determined by standard software quality metrics. Applicable research results are needed as soon as possible to facilitate the security evaluation process.

PANEL VIII BRIEFINGS - PART 6
ESTABLISH A MCCR SECURITY BASELINE FOR PDSS

1. A need exists to define a set of activities that should be performed to establish a security baseline for the PDSS of MCCRs. Such a baseline must exist in order to:

a. Judge the security impact of any changes to the MCCR.

b. Assure that changes do not degrade the security of the MCCR.

c. Allow for a defendable MCCR accreditation decision.

2. The set of activities must be accompanied by guidance and tools that streamlined and/or automated their execution. The activities must be generic enough to accommodate changing criteria and directives.

3. MCCRs exist in one of three states:

a. Deployed. systems that have been passed from development to either production or maintenance; considered operational in either case.

b. In-process. systems that have passed beyond the concept phase by virtue of an accepted Mission Element Needs Statement and funded program.

c. New starts. systems for which a demonstrated need has not been approved. States (a) and (b) above are similar in that they are both beyond the point where PDSS issues, related to MCCR security, must be considered.

4. Figure 6 depicts the activities needed to establish the security baseline and effectively perform PDSS for MCCR security. At stage one, a security requirements determination is made. Some work, such as NRL Report 8897 and various code audit tools, exists that apply to performing this activity. Assuming that a decision to proceed is made, stage two activities are conducted using the current accepted official security doctrines. Stage three establishes the security baseline through accreditation activities that permit effective PDSS for MCCR security.

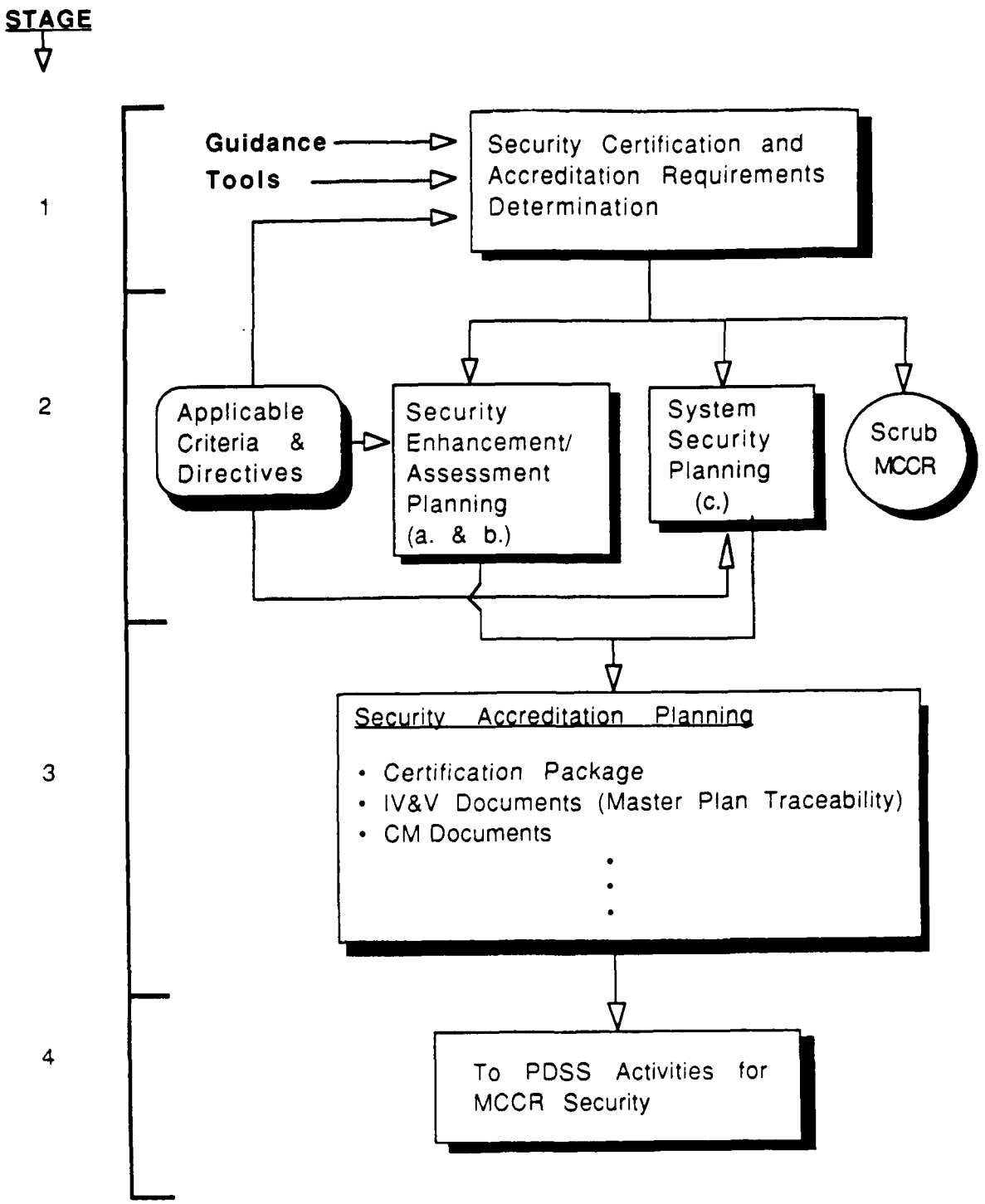


FIGURE 6. Creation of MCCR Security Baseline

PANEL VIII DAILY MINUTES
January 26, 1987

1. The MCCR Subpanel for Security in PDSS was kicked off with an introductory statement by each member. The panel membership included:

Robert A. Converse, Co-Chairman,	Computer Sciences Corp.
Sharon Muzik, Co-Chairman,	Naval Electronic System Engineering Activity
Charles A. B. Feldman	JASAR, Inc.
Bonnie Danner	LOGICON
David Imler	AFSC, Director, Information Systems Security
Glenn Meyers	AFSC, Directorate of Mission Critical Computer Resources
Michael Weidner	SYTEK
Jerry Cogar	AF Cryptologic Support Center
H. O. Lubbes	U.S. Navy Space and Warfare Systems Command
John Cole	U.S. Army Communications and Electronics Command

2. Mr. Lubbes briefed the genesis and intent of the Navy "Computer Security Acquisition Management Guidebook". He stated that the guidebook was initiated by OPNAV in 1984 in response to problems that arose in the acquisition of secure command and control systems. Until that time, security related guidelines had been primarily procedural and that technical guidance was needed to augment the procedures. Thus, the guidebook provides both technical guidance and procedural guidance for the development of secure systems and the configuration management of deployed secure systems. It also addresses the problems associated with cost/performance/benefit analysis as a means of determining the extent to which security assurance should be built into a system.

3. Mr. Feldman described an Integrated Configuration Environment concept that incorporates a software safety and hazard analysis, on which he has been working, that addresses the total system (hardware and software) issues associated with the support of (accredited) systems. Ms. Danner supported the importance of this concept and cited problems that arose on the Restricted Access Processor program developed for the National Aeronautics and Space Administration (NASA). The manual effort required to verify operational changes starting at the formal top level

design and flowing through implementation is extremely manpower intensive and would benefit greatly with the availability of some CM tools including an integrated data base.

4. There was general agreement that the development process associated with a secure system must be a highly disciplined process. The traditional method of top level design followed by ad hoc detailed designs developed by the programmers during implementation was grossly inadequate. This principal also was true for PDSS. System changes must be driven from the top level design rather than from the code by which it was implemented.

5. The issue of the definition of security was discussed at length. The consensus of the panel was that the definition was determined by the application area for the system to which security was to be applied. For information processing systems, a secure system is one that guarantees the integrity of and proper access to the information. For process oriented systems, such as a weapons control system, security means ensuring that the weapon will perform as intended (trustworthy - a safety issue); it is not inadvertently fired, it goes where it is supposed to, it is aimed correctly, and it is resistant to in-transit countermeasures. For control systems, such as a navigation system, the security aspect may be that the system always works (reliability). These examples identified (at least) two levels of abstraction for security:

(1) The integrity of the data on which application area actions are based.

(2) Assurance that the system will perform safely and reliably.

A failure in either information security or a process security situation can have disastrous results. If data integrity is not maintained, it can lead to the compromise of highly classified information or a contaminated data base which can cause an incorrect decision by a battlefield commander. If proper safeguards are not included in a weapons system, the platform from which it is fired can be destroyed, and, if a navigation system fails, the ship that depends on it can be hopelessly lost.

6. The preceding discussion demonstrated the need for security. The issue to be addressed by the panel is: how does this affect a PDSS Center? Four questions were posed:

a. How is a certified and accredited system affected over its life cycle?

b. How does an existing system obtain accreditation?

c. How does hardware maintenance for a secure system impact the PDSS Center?

d. How does a PDSS Center accomplish secure software maintenance and trusted distribution?

In order to answer these questions a description of certification and accreditation were supplied.

a. "Certification is the process of ensuring that an operational system precisely satisfies specified (security) criteria."

b. "Accreditation is a determination by the proper authority (Designated Approving Authority) that the operational system works well enough so that the operational need for the system outweighs the operational risk associated with system deployment when evaluated against the certification criteria."

7. In applying these definitions to operational systems, we determined that MCCS exist in one of three states:

a. Deployed.

b. In process.

c. New starts.

MCCR security activities for PDSS should be chosen in a manner that is independent of but takes into consideration applicable certification criteria. The set of activities to be applied is determined by the state of the MCCS. The variants of the set are the state of the system and the PDSS activities necessary to satisfy MCCR security requirements. The list of activities include:

a. Security certification and accreditation determination (all states).

b. Security enhancement assessment plan (deployed and in process states only).

c. Security accreditation plan (all states).

d. Security certification package (all states).

e. IV&V documentation.

8. Preliminary recommendations.

a. Strict systems and software engineering standards must be defined and enforced throughout the life cycle (development and post deployment) of the system.

b. Determine the DAA at the beginning and involve the DAA throughout the life cycle of a "secure" system.

c. Risk analysis and risk assessment must be a continuous process from requirements definition throughout the life. This includes verification and validation documentation for a deployed system to which security criteria are to be applied (after the fact).

d. Providing full IV&V documentation to the PDSS Center for a secure system is vital to the post deployment support process.

PANEL VIII DAILY MINUTES
January 27, 1987

1. Discussions were focused on the regulations, guidelines and policy directives associated with security and mission critical systems. The Panel's unanimous conclusion was that existing regulations and guidelines dealt with information processing security and did not adequately address the process security issues. Specific emphasis was placed on the Orange Book. The consensus was that the Orange Book provided certification criteria and requirements for operating systems that must support DOD security policy for application operations at a single level of classification as well as for those operating at more than one level of classification. The Orange Book does not provide a baseline for certifying or accrediting process control systems that require a different interpretation of the term "security". Also, even for operating systems, the guidance for application of the Orange Book does not recognize levels of granularity between the seven specified levels (D, C1, C2, B1, B2, B3 and A1). Systems may not require all of the security protection provided by an upper level, but may require more than is provided by the next lower level.

2. The Orange Book is a standard; however guidance for its use and application is inadequate. The Orange Book does provide a baseline for the derivation of criteria for other information processing security applications and for process control security applications. This baseline is provided by the theory and rationale contained in it, but other interpretations must be developed for data base, network, and process control application areas.

3. The next general topic for discussion was the support environment that exists in a PDSS center. This environment is usually a general purpose computer system whose resources are applied to the problem of providing life cycle support for a mission critical system. Such a system falls into the category of an information processing system for which the Orange Book criteria are applicable. However, further guidance is needed and the NRL Report entitled "An Approach to Determining Computer Security Requirements for Navy Systems", by Carl Landwehr and H. O. Lubbes, is an example of this guidance. The facilities that exist within a typical PDSS Center environment that supports the mission of the Center consist of:

a. Compiling (or language processing) capabilities.

b. Text preparation facilities.

c. Configuration management tools.

d. Test support tools.

e. Code analysis tools (perhaps).

f. Verification tools.

g. The operating system and run-time support tools.

Of these facilities, the operating system and run-time support tools are the only ones to which "rigorous" trusted computing criteria and methods must be applied; hence the applicability of the Orange Book. The other facilities are users of information protected by the operating system and are assumed to "do the proper thing" with that information. This assumption is usually shown to hold as a result of empirical data; they have worked properly for a long time.

4. Several specific writing assignments were made:

a. Charles Feldman will embellish the set of activities to be performed by a PDSS Center responsible for a MCCS that must meet security criteria.

b. David Imler will describe the role recommended for the DAA.

c. H. O. Lubbes will provide a detailed critique of the Orange Book.

d. Bonnie Danner will address security metrics.

e. John Cole will describe the PDSS facilities issues.

f. Michael Weidner will detail the R&D issues.

5. Preliminary recommendations.

a. Under the auspices of NSDD-145, the National Telecommunications and Information System and Security Committee must establish a single source for DOD computer security policy. The variety of existing DOD computer security policies and guidelines must be integrated into a single cohesive set, eliminating the confusion caused by conflicting direction.

b. A mechanism for assessing the impact of a change to a system on the security of that system must be defined. A necessary part is the placement of the DAA on the configuration control board for the system.

c. Orange Book requirements must be interpreted and guidance provided to address networks, data bases, and process control security applications as well as information systems security applications other than operating systems.

d. The JLC should establish a committee to develop changes to DOD-STD-2167 that incorporates security requirements as an integral part of a systems development life cycle. The standard must include specific Service requirements well as NCSC requirements; it must provide DIDs to detail the required deliverables; and it should be augmented by a guidebook for application of the security standards. (A starting point for such a standard is the "Computer Security Acquisition Management Guidebook, developed by SPAWAR.) The basis for this standard should proceed from an appropriate modification to DODD 5000.1 and DODD 5000.2.

e. The Agencies must be able to evaluate systems against trusted computing criteria and certify them for the accreditation process. (The certification process provided by the NCSC takes too long.)

f. The JLC should take steps to expedite the development and release of network certification criteria and data base certification.

g. The JLC should expedite the completion and release of standard language regarding security requirements for inclusion in contracts and SOW.

h. The Government must provide support for verification tools that are to be used for trusted systems development.

(Intentionally Blank)

PANEL VIII DAILY MINUTES
January 28, 1987

1. Discussions were centered on R&D issues. Dr. Michael Weidner focused on our efforts by providing a presentation on research topics that addressed the technology base, aids to development, and aids to evaluation. In the technology base area, he restated that there was a need for security oriented research for transitioning application to current technology areas such as networks, distributed systems, and data bases. He noted that hardware assisting in the support of security issues (e.g., for secure communications) were an ongoing effort as well as hardware support for data integrity within a specific architecture. He noted that security for the next generation of computing technologies (e.g., non-von Neumann architectures) was an issue that should be addressed now. Artificial intelligence technologies are sufficiently mature to be considered for application to security issues such as authentication, audit analysis, aggregation, resource control, penetration analysis, and (formal) verification support. Conversely, he noted that there is a security impact to the use of artificial intelligence in specific application areas such as communications, command, control, and intelligence. In the verification area, he noted that the integration of verification technologies into the software engineering disciplines to be applied to system development was a critical issue. In addition, he also stated that code verification is an ongoing effort. He further noted that the verification techniques that were applicable to software development could also be (and were being) applied to hardware development to achieve trusted results. He suggested some aids to the development process that could provide substantial benefits. Specifically, he described a security modeling capability that includes security model simulation and corresponding threat simulation, for which analysis tools could be provided to assess the extent to which security criteria were satisfied. He also suggested the development of software development tools that provides (semi) formal support for verification of the software from requirements analysis through code. This capability should include test generation and correspondence checks. Finally, to aid the evaluation process, he suggested a standard evaluation process (perhaps modeled after the TEMPEST process) and the development of evaluation tools such as penetration packages and automatic testing tools.

2. In applying this presentation to the PDSS problem, the panel identified R&D efforts from which near term, mid term, and long term benefits could accrue. In the near term (applicable to deployed and in development systems) the following efforts are suggested:

a. Security specific testing tools and methods that include penetration packages, regression test support, stress testing, and code analysis tools.

b. The definition and development of a standard evaluation process.

c. The adaptation and application of existing software engineering tools.

Those projects for which mid term (3 to 5 years) can be expected.

a. Security modeling for the solution space, the threat, and the necessary analysis.

b. The application of knowledge base technology to the automation of the software development process supporting the transition between development life cycle phases while preserving the complete traceability and providing (semi-) formal system verification.

Long term benefits can be expected from the hardware and architecture efforts and other undefined activities. The panel noted that mid and long term benefits would be realized for PDSS of new starts. Further, the panel noted that by including security requirements at the beginning of a project may allow the use of existing software tools to satisfy both security requirements as well as "conventional" software development requirements, thereby incurring little or no additional cost for security. Satisfying security requirements is really a rigorous application of a comprehensive quality assurance plan.

3. The discussion of metrics was limited to those that could be applied to the determination of the extent to which security requirements are met. Specific suggestions included:

a. The extent to which the (disciplined) development approach was followed.

b. The extent to which the specific security evaluation criteria are satisfied.

c. Based on the application of code analysis tools and techniques, an assessment of the code quality, the presence or amount of "dead" code, and the complexity of the code.

d. The extent to which the system is modularized and the degree to which security-critical code is isolated.

e. The anticipated amount of difficulty to accredit the system as a function of the perceived complexity of the system.

f. All standard software engineering quality metrics.

4. Preliminary recommendations.

a. Establish specific guidelines that address the security requirements for the transition of a system to a PDSS Center.

b. Establish a Security Efforts Coordination Agent under the JLC JPCG-CRM to make maximum use of individual Service security efforts.

(Intentionally Blank)

TABLE 6. MCCR Security Requirements Traceability

DOD-STD-5200.28 SECURITY REQUIREMENTS	AR 380-380	OPNAVINST 5239.1A	AFR 205-16	WWMCCS
Audit	9-4b	9.1	9a	nr
Configuration Management	9-2f	I.5b(4)	A6-9b	nr
Covert Channel Analysis	nr	nr	nr	nr
Design Documentation	13-1	nr	A6-6b	nr
Design Specification and Verification	nr	nr	nr	2.1.1.2
Device Labels	nr	nr	nr	nr
Discretionary Access Control	7-3n	J.3.41	A5-3/A5-3a/A5-3b	nr
Exportation of Labeled Information	nr	nr	nr	nr
Exportation to Multilevel Devices	nr	nr	nr	nr
Exportation to Single-Level Devices	nr	nr	nr	nr
Identification and Authentication	8-2e	J.3.4p(6)	9a	nr
Label Integrity	9-5b	J.3.4p(4)	A5-4	nr
Labeling Human-Readable Output	9-5b	J.3.4p(4)	A5-4	nr
Labels	nr	nr	nr	nr
Mandatory Access Control	7-3b/7-3m/8-2q/8-3e(3)	J.3.4b/J.3.414/J.3.4p3	9a/9e	nr
Object Reuse	9-5c	nr	A4-1	nr
Security Features User's Guide	nr	nr	nr	3-1
Security Testing	nr	6.1	pg 50	2.1.1.2
Subject Sensitivity Labels	nr	nr	nr	nr
System Architecture	ch 6/ch 7	J.3.4	nr	nr
System Integrity	nr	nr	nr	nr
Test Documentation	nr	nr	nr	2.1.1.2
Trusted Distribution	nr	nr	nr	nr
Trusted Facility Management	nr	nr	nr	nr
Trusted Facility Manual	nr	nr	nr	nr
Trusted Path	nr	nr	nr	nr
Trusted Recovery	9-3e(1)	J.3.4o	nr	nr

NR = Not a requirement of the document
Other references indicate chapter and paragraph

(Intentionally Blank)

PANEL VIII BIBLIOGRAPHY.
(SUGGESTED COMPUTER SECURITY RELATED READINGS)

1. Carl E. Landwehr and H. O. Lubbes, "An Approach to Determining Computer Security Requirements for Navy System", Naval Research Laboratory Report 8897, (Washington, D.C., May 1985).

2. Space and Naval Warfare Systems Command, "Computer Security Acquisition Management Guidebook", (Washington, D.C., 1986).

3. Micnael Schrage, "U.S. Limits Access to Information Related to National Security", Washington Post (Washington, D.C., November 1986).

4. National Telecommunications and Information Systems Security, NTISSP No. 2, "National Policy on Protection of Sensitive, but Unclassified Information in Federal Government Telecommunications and Automated Information Systems", (Washington, D.C., October 1986).

5. Department of Defense Directive 5200.28 (rewrite), "Security Requirements for Automated Information Systems (AISS)", (not dated).

6. Deborah J. Bodeau, "Security Models in the System Design and Development Process".

7. Terry S. Arnold, "Multilevel Security from a Practical Point of View".

8. Peter Gabriel Neumann, "On Hierarchical Design of Computer Systems for Critical Applications", IEEE Transactions on Software Engineering, (September 1986).

9. Donald I. Good, "The Foundations of Computer Security, We Need Some", (September 1986).

10. "Department of Defense Trusted Computer System Evaluation Criteria, (Orange Book)", December 1985, DOD 5200.28-STD.

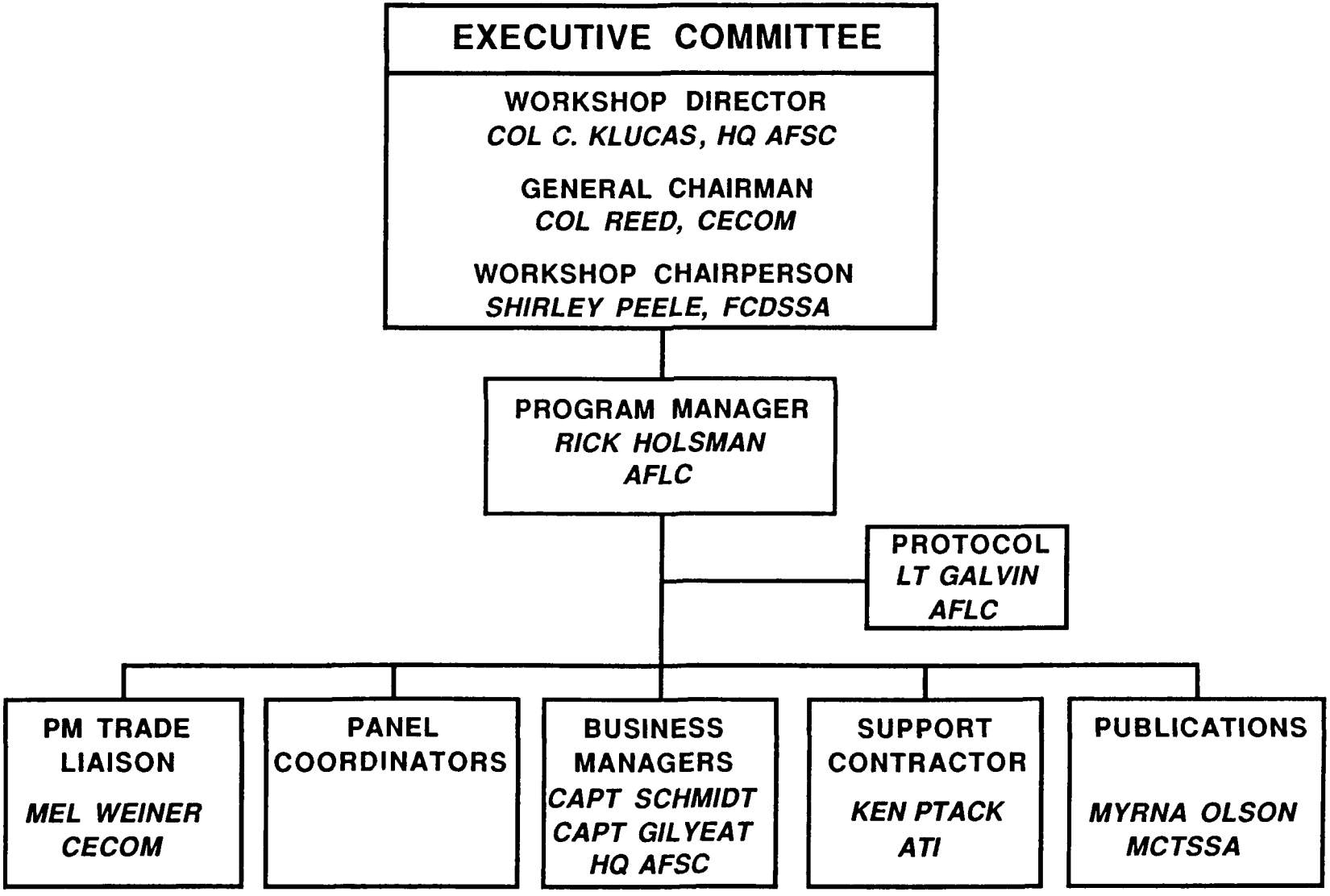
(Intentionally Blank)

APPENDIX A

WORKSHOP ORGANIZATION

(Intentionally Blank)

ORLANDO II
ORGANIZATION



APPENDIX B
ACRONYMS

ABET	Accreditation Board for Engineering Technology
ACQ	Acquisition
ADP	Automatic Data Processing
AFSC	Air Force Systems Command
AFSCP	Air Force Systems Command Pamphlet
AFLC	Air Force Logistics Command
AFR	Air Force Regulation
AMAT	Ada Measurement Analysis Tool
AMC	Army Materiel Command
AMS	Automated Measuring System
APSE	Ada Programming Support Environment
AR	Acquisition Regulation/Army Regulation
CAIS	Common APSE Interface Set
CAT	Complexity Analysis Tool
CCM	Configuration Control Management
CDR	Critical Design Review
CDRL	Contract Data Requirements List
C&E	Concept and Evaluation
CECOM	Army Communications-Electronics Command
CID	Configuration Identification Document
CLCSE	Centers for Life Cycle Software Engineering
CM	Configuration Management
CMP	Configuration Management Plan
CNO	Chief of Naval Operations
COTS	Commercial-Off-the-Shelf
COCOMO	Constructive Cost Model
CRISD	Computer Resources Interated Support Document
CRLCMP	Computer Resource Life Cycle Management Plan
CRMP	Computer Resource Management Plan
CRWG	Computer Resources Working Group
CSA	Configuration Status Accounting
CSAR	Configuration Status Accounting Report
CSCI	Computer Software Configuration Item
DA	Department of the Army
DAA	Designated Approving Authority
DAB	Defense Acquisition Board
DAR	Defense Acquisition Regulation
DCA	Defense Communications Agency
DCMC	DOD Configuration Management Committee
DCMP	DOD Configuration Management Program
DDMO	Defense Date Management Office
DFARS	Defense Federal Acquisiton Regulation Supplemental
DI	Data Item

DID	Data Item Description
DMSSO	Defense Material Specification and Standards Office
DNA	Defense Nuclear Agency
DOD	Department of Defense
DODI	Department of Defense Instruction
DODD	Department of Defense Directive
DOD-STD	Department of Defense Standard
DPSO	Defense Product Standards Office
DSAR	Defense Supply Agency Regulation
DSARC	Defense Systems Acquisition Review Council
DT&E	Development Test and Evaluation
D&V	Demonstration and Validation
ECP	Engineering Change Proposal
EIA	Electronic Industries Association
ERADCOM	U.S. Army Electronics Research Development Command
FAD	Force Activity Designator
FAR	Federal Acquisition Regulation
FASP	Facility for Automated Software Production
FCA	Functional Configuation Audit
FCDSSA	Fleet Combat Direction System Support Activity
FFRDC	Federally Funded Research Development Center
FQR	Formal Qualification Review
FW	Firmware
FY	Fiscal Year
GFA	Government Function Audit
GFE	Government Furnished Equipment
GFI	Government Furnished Information
GFS	Government Furnished Software
GS	General Schedule/Service
HDBK	Handbook
HOL	High Order Language
HQ	Headquarters
HWCI	Hardware Configuration Item
IAW	In Accordance With
ICWG	Interface Control Working Group
IDA	Institute for Defense Analysis
IEEE	Institute for Electrical and Electronic Engineers
IIT	Illinois Institute of Technology
IITRI	Illinois Institute of Technology Research Institute

ILS	Integrated Logistics Support
IRS	Interface Requirements Specification
ISC	Information Systems Command
ISEC	Information Systems Engineering Command
IV&V	Independent Verification and Validation
JLC	Joint Logistics Commanders
JPCG-CRM	Joint Policy Coordinating Group on Computer Resource Management
JPO	Joint Program Office
K	Thousand
KLOC	Thousand Lines of Code
LCC	Life Cycle Cost
LCM	Life Cycle Management
LOC	Lines of Code
MCCR	Mission Critical Computer Resources
MCCS	Mission Critical Computer Systems
MCDS	Mission Critical Defense Systems
MCO	Marine Corps Order
MCTSSA	Marine Corps Tactical Software Support Activity
MIL	Military
MIL-SPEC	Military Specification
MIL-STD	Military Standard
MSAT	Multistatic Analysis Tool
NADC	Naval Air Development Center
NASA	National Aviation and Space Administration
NASC	Naval Air Systems Command
NAV	Naval (or Navy)
NAVAIR	Naval Air Systems Command
NAVAIRINST	NAVAIRSYSCOM Instruction
NAVAIRSYSCOM	Naval Air Systems Command
NAVMATINST	Naval Material Command Instruction
NCSC	National Computer Security Center
NDI	Nondevelopmental Item
NDI/COTS	Nondevelopment Item/Commercial Off-the-Shelf
NMC	Naval Material Command
NOSC	Naval Ocean Systems Center
NRL	Naval Research Laboratory
NSA	National Security Agency
NTIS	National Telecommunications and Information Systems
NTISSC	National Telecommunications and Information Systems Security Committee

OM	Operator Manual
O&M	Operation and Maintenance
OMB	Office of the Management and Budget
OPM	Office of Personnel Management
OPNAV	Office of the Chief of Naval Operations
OPNAVINST	Office of the CNO Instruction
OPR	Office of Primary Responsibility
OS	Operating System
OSD	Office of the Secretary of Defense
OT	Operational Test
OT&E	Operational Test and Evaluation
PCA	Physical Configuration Audit
P&D	Product and Develcpment
PDA	Principal Development Activity or Preliminary Design Audit
PDD	Program Description Document
PDL	Program Design Language
PDR	Preliminary Design Review
PDSA	Post Deployment Support Activity
PDSS	Post Deployment Software Support
PDSSA	Post Deployment Software Support Activity
PID	Program Identification Document
PM	Program Manager
PMTC	Pacific Missile Test Center
PPBS	Planning Programming and Budgeting System
PRR	Production Readiness Review
PSE	Program Support Environment
QA	Quality Assurance
QAP	Quality Assurance Plan
RADC	Rome Air Development Center
R&D	Research and Development
RFP	Request for Proposal
ROI	Return On Investment
SAIC	Science Applications International Corporation
SCCAS	Software Change Control Automation System
SCCB	Software Configuration Control Board
SCE	Software Cost Estimating
SCMP	Software Configuration Management Plan
SCN	Software Change Notice
SCO	Software Change Order
SCP	Software Change Proposal
SCRB	Software Configuration Review Board
SDE	Software Development Environment
SDF	Software Development File (Folder)
SDP	System Development Plan or Software Development Plan

SDR	System Design Review
SECOMO	Software Engineering Cost Model
SEE	Software Engineering Environment
SEI	Software Engineering Institute
SIE	System Integration Environment
SLP	Software Licensing Project
SOW	Statement of Work
SPAR	Source Program Analyzer and Reporter
SPAWAR	Space and Naval Warfare Systems Command
SPEC	Specification
SPO	Standardization Program Office
SRR	System Requirements Review
SRS	System Requirements Specification
SQAP	Software Quality Assurance Plan
SQEP	Software Quality Evaluation Plan
SQPP	Software Quality Program Plan
SSE	Software Support Environment
SSPM	Software Standards and Practices Manual
SSR	Software Specifications Review or Software Support Review
SSRA	Software Support Requirements Analysis
SSS	System/Segment Specification
SSSA	Systems Software Support Activity
STARS	Software Technology for Adaptable and Reliable Systems
STD	Standard/Software Test Description
STP	Software Test Plan or Software Test Procedures
STPR	Software Test Plan Report
STR	Software Test Report or Software Trouble Report
TCB	Trusted Computer Base
TDS	Tactical Data Systems
TECR	Tactical Embedded Computer Resources
TEMP	Test and Evaluation Master Plan
TR	Technical Review
TRADOC	Army Training and Doctrine Command
TRR	Test Readiness Review
USA	United States Army
USAF	United States Air Force
USMC	United States Marine Corps
USN	United States Navy
VDD	Version Description Document
VHSIC	Very High Speed Integrated Circuit
WBS	Work Breakdown Structure

APPENDIX C
JPCG-CRM MEMBERS

CRM PRIMARY MEMBERS

HQ AFSC/PLR (CHR CRM)
Attn: Casper Klucas, Col
Andrews AFB, MD 20334-5000

SPAWAR (Code 3212)
Attn: Gary Taul, CDR
(CP 6), Room 684
Washington, DC 20363-5100

HQ AMC/AMCDE-SB
Attn: Victor Shavers, Col
5001 Eisenhower Ave.
Alexandria, VA 22333-0001

HQ AFLC/MMT
Attn: Mark Burroughs, Col
Wright-Patterson AFB,
OH 45433-5001

HQ US Marine Corps (Code CCA-51)
Attn: Mark Stotzer, Capt
Washington, DC 20380-0001

CRM ALTERNATE MEMBERS

HQ AFSC/PLR
Attn: Rick Butler, Maj
Andrews AFB, MD 20334-5000

SPAWAR (Code 321B)
Attn: John Machado
(CP 6), Room 944
Washington, DC 20363-5100

HQ AMC/AMCDE-SB-S
Attn: Bob Loomis, Dr.
5001 Eisenhower Ave. Rm 9M-13
Alexandria, VA 22333-0001

HQ AFLC/MMTEC
Attn: Randy Adams, LtCol
Wright-Patterson AFB,
OH 45433-5001

HQ US Marine Corps (CCA-50)
Attn: Heinz McArthur, Capt
Washington, DC 20380-0001

APPENDIX D
PDSS SUBGROUP MEMBERS

PDSS SUBGROUP PRIMARY MEMBERS	PDSS SUBGROUP ALTERNATE MEMBERS
FCDSSA (Code 82), Dam Neck Attn: Shirley Peele (CHR PDSS) Va. Beach, VA 23461	PMTC, Code 4023 Attn: Lucille Cook Point Mugu, CA 93042
MCTSSA/CC-QA Attn: Ron Pruiett, LtCol Camp Pendleton, CA 92055-5080	MCTSSA/CC-QA Attn: Myrna Olson Camp Pendleton, CA 92055-5080
U.S. Army CECOM/AMSEL-RD-LC-SPM3 Attn: Mel Weiner Ft. Monmouth, NJ 07703-5300	US Army CECOM LCSE Center Attn: Joseph J. Potoczniak AMSEL-SDSC-SD Ft. Monmouth, NJ 07703-5300
HQ AFSC/PLRP Attn: Colin Gilyeat, Capt Andrews AFB, MD 20334-5000	HQ AFSC/PLRP Attn: Richard F. Schmidt, Capt Andrews AFB, MD 20334-5000
OO-ALC/MMEC Attn: Eldon Jensen Hill AFB, UT 84056	OO-ALC/MMEC Attn: Rick Holsman Hill AFB, UT 84056

APPENDIX E

ALPHABETICAL LIST OF ATTENDEES

ALPHABETICAL LIST OF ATTENDEES

NAME	ORGANIZATION	PANEL
Adams, Maj Randy	HQ Air Force Logistics Command	Staff
Armour, Capt Rich	HQ USAF	I
Baker, Dr. Emanuel	Software Engineering Consultants	VII
Bates, Wayne	Ogden Air Logistics Center	VII
Bausman, Karen	ASD	IV
Baxter, Bruce	Pacific Missile Test Center	I
Bedar, Maj George	MCTSSA	VII
Benson, John	Bell Helicopter	III
Berlack, Ronald	Sanders Associates	III
Boehm, Barry	TRW	II
Both, Robert	Army Communications Electronics Command	III
Bracker, Lynne	Hughes Aircraft	VII
Branyan, Elmer	General Electric	II
Brim, Terry	TRW	VI
Bornako, Greg	FCDSSA, Dam Neck	IV
Brooks, Sharon	AD	VI
Burroughs, Col Mark	HQ Air Force Logistics Command	Staff
Butler, Maj Rick	HQ Air Force Systems Command	IV
Byerly, Paul	Naval Training Systems Center	IV
Byers, Jack	HQ Army Material Command	I
Calland, Robert	Naval Ocean Systems Center	VII
Castellano, David	AMCCOM	IV
Cavano, Joe	Rome Air Development Center	V
Cogar, Capt Gerald	AFCSC	VIII
Cole, John	Army Communications-Electronics Command	VIII
Collier, Linda	MCTSSA	V
Conrad, Thomas	Naval Underwater Systems Center	IV
Converse, Bob	Computer Sciences Corporation	VIII
Cook, Lucille	Pacific Missile Test Center	VI
Cooper, Jack	Anchor Software Management	VII
Cooper, Lee	Advanced Technology	V
Corson, Barry	Naval Air Systems Command	V
Cover, Donna	IIT Research Institute	I
Cruickshank, Robert	IBM	III
Curtis, Col Lewis	HQ Air Force Logistics Command	Staff
Dada, Cenap	Army Communications-Electronics Command	I
Danner, Bonnie	Logicon	VIII
Davidson, Capt Charles	AD	V
Davis, Paula	OPNAV	I
Day, Raymond	Intercon Systems	V
DeWeese, Perry	Lockheed Georgia	III
Doldt, Linda	HQ Air Force Logistics Command	VI
Edgerton, Russell	AFALC	I
Egan, Bill	Advanced Technology	I
Feldman, Charles	JASAR	VIII

Fisher, Dr. Matt	Army Communications-Electronics Command	V
Floyd, Jon	General Dynamics	III
Fowler, Priscilla	Software Engineering Institute	VI
Frogner, Don	Sacramento Air Logistics Center	I
Galvin, 1st Lt Linda	Ogden Air Logistics Center	Staff
Gant, Donna	General Dynamics	V
Gilyeat, Capt Colin	HQ Air Force Systems Command	Staff
Gladson, Clell	Naval Ocean Systems Center	V
Glushko, Robert	Software Engineering Institute	VII
Goethert, Wolf	IIT Research Institute	II
Golubjatnikov, Ole	General Electric	IV
Goudy, Ron	MCTSSA	II
Green, Dan	Naval Surface Weapons Center	VI
Hansen, Gregory	Software Engineering Institute	III
Harris, Andrew	HQ USMC	VI
Harvey, Lawrence	Teledyne Brown Engineering	VII
Hatakeyama, Kris	NSWSES	III
Havey, Robert	Defense Technology Analysis Office	III
Healy, Richard	Atlantis Research Group	I
Heil, James	ITT Avionics	IV
Holcomb, John	Oklahoma City Air Logistics Center	I
Holinko, Myron	Army Communications-Electronics Command	VII
Hubans, Frank	General Dynamics	III
Imler, David	HQ Air Force Systems Command	VIII
Ipsen, Karl	Army Communications-Electronics Command	VI
Irwin, Allen	SAIC	VII
Jankowski, Cheryl	Army Communications-Electronics Command	II
Janusz, Paul	AMCCOM	V
Kelly, Charles	FCDSSA, Dam Neck	IV
Klucas, Col Casper	HQ Air Force Systems Command	Staff
Kluge, Clyde	Oklahoma City Air Logistics Center	III
Knutson, LtCol Darrel	AFOTEC	V
Koch, Charles	Naval Air Development Center	V
Krabbe, Kurt	TRW	IV
Kvenvold, Dan	HQ Air Force Logistics Command	IV
Leask, Ron	Naval Underwater Systems Center	II
Lee, Kenneth	Army Communications-Electronics Command	I
Lipke, Walter	Oklahoma City Air Logistics Center	VI
Lohr, Claire	Lohr Software	III
Long, Gene	Sacramento Air Logistics Center	V
Lubbes, H.O.	SPAWAR	VIII
Maibor, David	Dynamics Research Corporation	IV
Malinowski, Gregory	Army Communications-Electronics Command	VII
Marciniak, John	Marciniak and Associates	VII
Martin, Anne	Software Engineering Institute	I
Mauro, Paul	Hughes Aircraft	I
McCall, Jim	SAIC	V
McCormick, Bob	HQ Air Force Logistics Command	II
McDonald, James	AFWAL	VII
McOmber, Owen	Comptek Research	III
Mendis, Ken	Raytheon	VI
Merry, Hubert	Ogden Air Logistics Center	V
Meyers, Capt Glenn	HQ Air Force Systems Command	VIII
Miller, Jim	SAIC	V

Murray, William	General Dynamics	VII
Muzik, Sharon	NESEA	VIII
Nickle, Dennis	E-Systems	III
Nidiffer, Ken	Software Productivity Consortium	II
Norton, Henry	Pacific Missile Test Center	III
Nuhn, Perry	Software Productivity Consortium	VII
Oglesby, Charles	HQ Army Material Command	VI
Osborne, Wilma	National Bureau of Standards	III
Packer, Stanley	Ogden Air Logistics Center	IV
Pariseau, Richard	NCSC	III
Parlier, Jim	General Dynamics	IV
Parsley, Vern	Computer Sciences Corporation	IV
Perkins, John	Dynamics Research Corporation	V
Pollard, LtCol Ray	MCDEC	II
Porter, Kevin	Control Data Corporation	I
Potter, Marshall	OPNAV	II
Preston, David	IIT Research Institute	VII
Price, Bernie	Army Communications-Electronics Command	II
Pruiett, LtCol Ron	MCTSSA	III
Przybylinski, Stan	Software Engineering Institute	VI
Ptack, Ken	Advanced Technology	Staff
Radatz, Jane	Logicon	IV
Raveling, Jerry	Unisys	II
Reed, Col Scott	Army Communications-Electronics Command	Staff
Reichson, Jack	Honeywell Electro Optics	IV
Rhoads, Dean	Unisys	III
Rodriquez, Albert	Army Communications-Electronics Command	VII
Romero, Capt Tony	MCTSSA	III
Ruckstuhl, John	Naval Training Systems Center	III
Sanders, Linda	MCTSSA	I
Santo-Donato, Arthur	Army Communications-Electronics Command	VI
Shavers, Col Victor	HQ Army Material Command	Staff
Sherer, S. Wayne	ARDEC	IV
Shumskas, Maj Tony	HQ Air Force Systems Command	V
Simmons, Capt Denise	MCTSSA	VI
Singh, Dr. Raghu	SPAWAR	V
Sisti, Francis	Vitro	I
Skullman, Victor	Naval Air Systems Command	III
Smith, Jerry	QSOFT	VII
Sonnenblick, Paul	Expertware	I
Soskins, Frances	Telos Corporation	I
Spaulding, William	Dynamics Research Corporation	I
Steenwerth, James	MCTSSA	IV
Stees, Mae	Mae Stees and Associates	IV
Sterling, Jack	Army Communications-Electronics Command	II
Stewart, Col James	MCTSSA	I
Stewart, Lee	Army Missile Command	IV
Stewart, Marilyn	Logicon	I
Stuebing, Hank	Naval Air Development Center	II
Szymanski, Raymond	AFWAL	V
Taul, Cdr Gary	SPAWAR	Staff
Wagner, James	Army Communications-Electronics Command	I
Wasgatt, Bud	NSWSES	I
Wasilausky, Robert	Naval Ocean Systems Center	VII

Weidner, Michael	SYTEK	VIII
Westaway, Thomas	Sacramento Air Logistics Center	II
Whitley, Lee	Telos Corporation	II
Winter, Milt	Army Communications-Electronics Command	II
Wood, Dennis	Software Enterprises Corporation	II
Zana, Don	Teledyne Brown Engineering	I

APPENDIX F

ATTENDEES ADDRESS LIST

STAFF MEMBERS

Post Deployment Software Support Subgroup

Holsman, Mr. Rick
OO-ALC/MMEC
Hill AFB, UT 84056
(801) 777-7355
A/V 458-7355

Olson, Ms. Myrna
MCTSSA/CC-QA
Camp Pendleton, CA 92055-5080
(619) 725-2502
A/V 365-2502

Peele, Ms. Shirley
Code 82
FCDSSA, Dam Neck
Virginia Beach, VA 23461
(804) 433-7257
A/V 433-7257

Schmidt, Capt Richard
HQ AFSC/PLRP
Andrews AFB, MD 20334-5000
(301) 981-5731
A/V 858-5731

Weiner, Mr. Mel
CECOM
AMSEL-RD-LC-SPM3
Fort Monmouth, NJ 07703-5000
(201) 532-4280
A/V 992-4280

Joint Policy Coordinating Group
On
Computer Resource Management

Adams, Maj Randy
HQ AFLC/TEC
Wright-Patterson AFB, OH 45433-5001
(513) 257-2056
A/V 787-2056

Burroughs, Col Mark
HQ AFLC/MME
Wright-Patterson AFB, OH 45433-5001
(513) 257-2258
A/V 787-2258

Klucas, Col Casper
HQ AFSC/PLR
Andrews AFB, MD 20334-5000
(301) 981-5731
A/V 858-5731

Shavers, Col Victor
HQ AMC/AMCDE-SB
5001 Eisenhower Ave.
Alexandria, VA 22333-0001
(703) 274-9310
A/V 284-9310

Taul, Cdr Gary
SPAWAR, Code 3212
Washington, DC 20363-5100
(202) 692-9097
A/V 222-9097

Computer Software Management Subgroup

Gilyeat, Capt Colin
HQ AFSC/PLRP
Andrews AFB, MD 20334-5000
(301) 981-5731
A/V 858-5731

Other Staff Members

Curtis, Col Lewis
HQ AFLC/MM
Wright-Patterson AFB, OH 45433
(513) 257-3022
A/V 787-3022

Galvin, 1st Lt Linda
OO-ALC/MMEAR
Hill AFB, UT 84056
(801) 777-7565
A/V 458-7565

Ptack, Mr. Ken
Advanced Technology, Inc.
Suite 1211
1235 Jefferson Davis Hwy.
Arlington, VA 22202
(703) 892-0900

Reed, Col Scott
CECOM
AMSEL-RD-LC-DD
Fort Monmouth, NJ 07703-5300
(201) 544-4211
A/V 995-4211

Panel I: PDSS Planning During Development

Government

Co-Chairman

Holcomb, Mr. John
OC-ALC/MMECM
Tinker AFB, OK 73145-5990
(405) 736-5609
A/V 336-5609

Armour, Capt Rich
HQ USAF/SCPX
Washington, DC 20330-5190
(202) 695-0756
A/V 225-0756

Baxter, Mr. Bruce
Code 4020
Pacific Missile Test Center
Point Mugu, CA 93042
(905) 989-9405
A/V 351-9405

Byers, Mr. Jack
HQ AMC
AMCDE-SB-S
5001 Eisenhower Ave.
Alexandria, VA 22333-0001
(202) 274-9309
A/V 284-9309

Dada, Mr. Cenap
CECOM
AMSEL-RD-LC-IEW-1B
Ft. Monmouth, NJ 07703-5000
(201) 544-2291
A/V 995-2291

Davis, Ms. Paula
OPNAV-945C
CNO, Pentagon
Washington, DC 20350-2000
(202) 697-7216
A/V 227-7216

Edgerton, Mr. Russell D.
AFALC/EREC
Wright Patterson AFB, OH 45433-5000
(513) 255-4991
A/V 785-4991

Frogner, Mr. Don
SM-ALC/MMEC
McClellan AFB, CA 95611
(916) 643-6454
A/V 633-6454

Lee, Mr. Kenneth
CECOM
AMSEL-RD-LC-SPM-2A
Ft. Monmouth, NJ 07703-5301
(201) 544-4791
A/V 995-4791

Martin, Ms. Anne
Software Engineering Institute
Carnegie-Mellon University
580 South Aiken
Pittsburgh, PA 15213
(412) 268-7788

Sanders, Ms. Linda J.
MCTSSA
Camp Pendleton, CA 92055-5080
(619) 725-2721
A/V 365-2721

Stewart, Col James J.
MCTSSA
Camp Pendleton, CA 92055-5080
(619) 725-2618
A/V 365-2618

Wagner, Mr. James R.
CECOM
AMSEL-RD-LC-COM
Fort Monmouth, NJ 07703-5000
(201) 532-5848
A/V 992-5848

Wasgatt, Mr. Bud
Code 4L20
NSWSES
Port Hueneme, CA 93030
(805) 982-3751
A/V 360-3751

Panel I: PDSS Planning During Development

Industry

Co-Chairman

Egan, Mr. Bill
Advanced Technology, Inc.
751 Daily Dr., Suite 220
Camarillo, CA 93010
(805) 987-8831

Cover, Ms. Donna
IIT Research Institute
Suite 300
4550 Forbes Blvd.
Lanham, MD 20706-4324
(301) 459-3711

Healy, Mr. Richard D.
Atlantis Research Group
1 Intercontinental Way
Peabody, MA 01960
(617) 535-4747

Mauro, Mr. Paul
Hughes Aircraft Co.
Bldg. 618/B218
P.O. Box 3310
Fullerton, CA 92634
(714) 732-4052

Porter, Mr. Kevin E.
Control Data Corporation
60 Hickory Drive
Waltham, MA 02154
(617) 466-6480

Sisti, Mr. Francis J.
Vitro Corporation
Suite A609
1111 Army Navy Dr.
Arlington, VA 22202
(703) 553-8245

Sonnenblick, Mr. Paul
Expertware, Inc.
Suite 1209
2685 Marine Way
Mountain View, CA 94043
(415) 965-8921

Soskins, Ms. Frances
Telos Federal Systems
Suite 3050
3420 Ocean Park Blvd.
Santa Monica, CA 90405
(213) 450-2424

Spaulding, Mr. William J.
Dynamics Research Corporation
60 Frontage Road
Andover, MA 01810
(617) 475-9090

Stewart, Ms. Marilyn
Logicon Suite 1000
1815 N. Lynn St
Arlington, VA 22209
(703) 243-6606

Zana, Mr. Don
Teledyne Brown Engineering
MS166
300 Sparkman Drive
Huntsville, AL 35807
(205) 876-9388

Panel II: Forecasting PDSS Resources Requirements

Government

Co-Chairman

Price, Mr. Bernie
CECOM
AMSEL-PL-SA
Fort Monmouth, NJ 07703-5000
(201) 532-1222
A/V 992-1222

Goudy, Mr. Ron
MCTSSA
Camp Pendleton, CA 92055-5080
(619) 725-2618
A/V 365-2618

Jankowski, Ms. Cheryl
CECOM
AMSEL-RD-LC-SPM-3D
Fort Monmouth, NJ 07703-5000
(201) 532-4280
A/V 992-4280

Leask, Mr. Ron
Code 2153
NUSC
New London, CT 06320
(203) 440-4366
A/V 241-4366

McCormick, Mr. Robert
HQ AFLC/ACCCE
Wright Patterson AFB, OH 45433-5001
(513) 257-3920
A/V 787-3920

Pollard, LtCol Ray
Code D04
Marine Corps Development Center
Quantico, VA 22134-5080
(703) 640-2547
A/V 278-2547

Potter, Mr. Marshall
OASN (FM) DIRDONIRM
Pentagon, RM 4E768
Washington, DC 20350
(202) 697-9346
A/V 227-9346

Sterling, Mr. Jack
CECOM
AMSEL-CP-CA
Fort Monmouth, NJ 07703-5000
(201) 544-4119
A/V 995-4119

Stuebing, Mr. Hank
Code 70C
NADC
Warminster, PA 18974-5000
(215) 441-2314
A/V 441-2314

Westaway, Mr. Thomas A.
SM-ALC/MMARA
McClellan AFB, CA 95652-5609
(916) 643-6388
A/V 633-6388

Winter, Mr. Milt
CECOM
AMSEL-PL-SA
Fort Monmouth, NJ 07703-5000
(201) 532-1222
A/V 992-1222

Panel II: Forecasting PDSS Resources Requirements

Industry

Co-Chairman

Raveling, Mr. Jerry
Unisys
Computer Systems Division
P.O. Box 64525
MS CD2D04
St. Paul, MN 55164-0525
(612) 681-6800

Boehm, Mr. Barry
TRW, MS R2-2086
1 Space Park
Rendondo Beach, CA 90278
(213) 535-2184

Branyan, Mr. Elmer
General Electric Co.
P.O. Box 8048
Room 10858
Philadelphia, PA 19101
(215) 531-1001

Goethert, Mr. Wolf
IIT Research Institute
Turin Road North
P.O. Box 180
Rome, NY 13440
(315) 336-2359

Nidiffer, Mr. Ken
Software Productivity
Consortium, Inc.
1880 N. Campus Drive
Reston, VA 22091
(703) 391-1820

Whitley, Mr. Lee
Telos Corporation
711 Southwest Avenue D
Lawton, OK 73501
(405) 355-9280

Wood, Mr. Dennis
Software Enterprises Corp.
Suite 110
31220 La Baya Drive
Westlake Village, CA 91362
(818) 889-7814

Panel III: Software Change Process

Government

Co-Chairman

Pruiett, LtCol Ron
MCTSSA
Camp Pendleton, CA 92055-5080
(619) 725-2618
A/V 365-2618

Both, Mr. Robert J.
CECOM
AMSEL-RD-LC-SPM-3A
Ft. Monmouth, NJ 07703
(201) 532-1898
A/V 992-1898

Hansen, Mr. Gregory
Software Engineering Institute
Carnegie-Mellon University
580 South Aiken
Pittsburgh, PA 15232
(412) 268-7622

Hatakeyama, Mr. Kris
Code 4L13
NSWSES
Port Hueneme, CA 93043
(805) 982-3751
A/V 360-3751

Havey, Mr. Robert
DOD Technology Analysis Office
1221 South Fern, Room C-107
Arlington, VA 22202
(202) 694-0865
A/V 224-0865

Kluge, Mr. Clyde
OC-ALC/MMECT
Tinker AFB, OK 73145
(405) 736-5700
A/V 336-5700

Norton, Mr. Henry J.
Code 02-A
PMTC
Point Mugu, CA 93042
(805) 989-7202
A/V 351-7202

Osborne, Ms. Wilma
Bldg 225, Room 266
NBS, ICST
Washington, DC 20234
(301) 921-3545

Pariseau, Mr. Richard
Code 30D
NCSC
Panama City, FL 32407
(904) 234-4113
A/V 436-4113

Romero, Capt Tony
MCTSSA
Camp Pendleton, CA 92055
(619) 725-2421
A/V 365-2421

Ruckstuhl, Mr. John
Code 253
NTSC
Orlando, FL 32813
(305) 273-4891
A/V 791-4111

Skullman, Mr. Victor
Code 54661
NAVAIR
Washington, DC 20361-0546
(202) 692-6226
A/V 222-6226

Panel III: Software Change Process

Industry

Co-Chairman

McOmber, Mr. Owen
Comptek Research, Inc.
2929 Canon St, Suite 200
San Diego, CA 92106
(619) 225-9921

Benson, Mr. John
Airborne Software
MS6, Dept. 81
Bell Helicopter
P.O. Box 482
Ft. Worth, TX 76101
(817) 280-5042

Berlack, Mr. Ronald
Sanders Associates NCA1-3286
P.O. Box CS2004
Nashua, NH 03061
(603) 885-5170

Cruickshank, Mr. Robert D.
IBM, Federal Systems Division
Building 895
9500 Goodwin Drive
Manassas, VA 22110
(703) 367-3258

DeWeese, Mr. Perry
Bldg. L10, Zone 410, Dept. 7282
Lockheed Georgia
86 South Cobb Drive
Marietta, GA 30063
(404) 425-6198

Floyd, Mr. Jon
Mail Zone 1880
General Dynamics
P.O. Box 748
Fort Worth, TX 76101
(817) 777-4416

Hubans, Mr. Frank
Mail Zone 1783
General Dynamics
P.O. Box 748
Ft Worth, TX 76101
(817) 777-6474

Lohr, Ms. Claire L.
Lohr Software Systems Corp.
2915 Hunter Mill Rd., Suite 6
Oakton, VA 22124
(703) 281-5553

Nickle, Mr. Dennis
Melpar Division, E-Systems
7700 Arlington Blvd.
Falls Church, VA 22046
(703) 849-1559

Rhoads, Mr. Dean I.
Unisys
10036 Scenic View Terrace
Vienna, VA 22180
(703) 620-7999

Panel IV: PDSS Standards

Government

Co-Chairman

Packer, Mr. Stanley
OO-ALC/MMGRA
Hill AFB, UT 84056-5609
(801) 777-1787
A/V 458-1787

Bausman, Ms. Karen
ASD/ENST
Wright-Patterson AFB, OH 45433-6503
(513) 255-3500
A/V 992-5759

Bornako, Mr. Gregory
FCDSSA, DN, Code 6216
Virginia Beach, VA 23461-5300
(804) 433-7639
A/V 433-7639

Butler, Maj Rick
HQ AFSC/PLPR
Andrews AFB, MD 20334-5000
(301) 981-5731
A/V 858-5731

Byerly, Mr. Paul
NTSC, Code 251
Orlando, FL 32813-7100
(305) 646-5354
A/V 791-5354

Castellano, Mr. David R.
AMCCOM
AMSMC-QAH-A(D)
Bldg 62
Dover, NJ 07801-5001
(201) 724-2305
A/V 880-2305

Conrad, Mr. Thomas P.
Code 2211, Bldg 1171/1
Naval Underwater Systems Center
Newport, RI 02841-5047
(401) 841-3354
A/V 948-3354

Kelly, Mr. Charles
FCDSSA, DN, Code 212
Virginia Beach, VA 23461-5300
(804) 433-7306
A/V 433-7306

Kvenvold, Mr. Dan
HQ AFLC/MMTEC
Wright Patterson AFB, OH 45433-5001
(513) 257-6753
A/V 787-6753

Sherer, Mr. S. Wayne
Armament Research Development
Engineering Center
SMCAR-FSC, Bldg 352
Dover, NJ 07801-5001
(201) 724-3531
A/V 880-3531

Steenwerth, Mr. James
MCTSSA
Camp Pendleton, CA 92055
(619) 725-2907
A/V 365-2907

Stewart, Mr. Lee
U.S. Army Missile Command
AMSMI-RD-BA-SE-CT
Redstone Arsenal, AL 35898-5091
(205) 876-4442
A/V 746-4442

Panel IV: PDSS Standards

Industry

Co-Chairman

Parsley, Mr. Vern
Computer Sciences Corporation
4045 Hancock Street
San Diego, CA 92110
(619) 225-8401

Golubjatnikov, Mr. Ole
Plant 1, Room D6
General Electric Co.
Farrell Road
Syracuse, NY 13221
(315) 456-4744

Heil, Mr. James
ITT Avionics-74201
100 Kingsland Road
Clifton, NJ 07014
(201) 284-2946

Krabbe, Mr. Kurt
HSV 08/11
TRW/DSG
213 Wynn Drive
Huntsville, AL 35805
(205) 721-3105

Maibor, Mr. David
Dynamics Research Corporation
60 Frontage Road
Andover, MA 01810
(617) 475-9090

Parlier, Mr. Jim
MZ V-2 5530
Data Systems Division
General Dynamics
P.O. Box 85808
San Diego, CA 92138
(619) 573-3747

Radatz, Ms. Jane
Logicon
4010 Sorrento Vally Blvd.
P.O. Box 85158
San Deigo, CA 92138-5158
(619) 455-1330

Reichson, Mr. Jack M.
Mail Stop 355
Honeywell Electro-Optics
#2 Forbes Road
Lexington, MA 02173
(617) 863-3907

Stees, Ms. Mae
Mae Stees and Associates
P.O. Box 2775
Costa Mesa, CA 92628
(714) 545-7993

Panel V: PDSS Management Indicators and Metrics

Government

Co-Chairman

Long, Mr. Gene
SM-ALC/MMEA
McClellan AFB, CA 95652-5609
(916) 643-4198
A/V 633-4198

Cavano, Mr. Joe
RADC/COEE
Griffiss AFB, NY 13441
(315) 330-4063
A/V 587-4063

Collier, Ms. Linda
MCTSSA
Camp Pendleton, CA 92055-5080
(619) 725-2415
A/V 365-2415

Corson, Mr. Barry A.
Code A5466
NAVAIR
Washington, DC 20361
(202) 692-6226
A/V 286-6226

Davidson, Capt Charles E.
HQ AD/ENEC
Eglin AFB, FL 32542-5000
(904) 882-8505
A/V 872-8505

Fisher, Dr. Matthew J.
CECOM
AMSEL-RD-LC-IEW
Fort Monmouth, NJ 07703
(201) 544-4741
A/V 995-4741

Gladson, Mr. Clell
NOSC, Code 9201
271 Catalina Blvd.
San Diego, CA 92152-5000
(619) 225-7615
A/V 933-7615

Janusz, Mr. Paul E.
AMCCOM
AMSMC-QAH-A(D) Bldg. 62
Dover, NJ 07801-5001
(210) 724-2305
A/V 880-2305

Knutson, LtCol Darrel
AFOTEC/LG5
Kirtland AFB, NM 87117-7001
(505) 846-1259
A/V 246-1259

Koch, Mr. Charles F.
Code 709C
Naval Air Development Center
Warminster, PA 19874-5000
(215) 441-3794
A/V 441-3794

Merry, Mr. Hubert
OO-ALC/MMFRC
Hill AFB, UT 84056-5609
(801) 777-7542
A/V 458-7542

Shumskas, Maj Tony
HQ AFSC/PLRP
Andrews AFB, MD 20334-5000
(301) 981-5731
A/V 858-5731

Singh, Dr. Raghu
SPAWAR, Code 321
CP6, Room 944
Washington, DC 20363-5100
(202) 692-9207
A/V 222-9207

Szymanski, Mr. Raymond
AFWAL/AAAF-2
Wright Patterson AFB, OH 45433-6543
(513) 255-2446
A/V 785-2446

Panel V: PDSS Management Indicators and Quality Metrics

Industry

Co-Chairman

Miller, Mr. Jim
SAIC
2815 Camino Del Rio, South
San Diego, CA 92108
(619) 293-7500

Cooper, Mr. Lee
Advanced Technology, Inc.
Suite 1211
1235 Jefferson Davis Hwy.
Arlington, VA 22202
(703) 892-0900

Day, Mr. Raymond
Intercon Systems Corp.
9400 Viscount, Suite 115
El Paso, TX 79925
(915) 593-5043

Gant, Ms. Donna
General Dynamics Data Systems
12101 Woodcrest Executive Drive
St. Louis, MO 63141
(314) 851-8991

McCall, Mr. Jim
SAIC
P.O. Box 2351
La Jolla, CA 92038
(619) 456-6220

Perkins, Mr. John
Dynamics Research Corporation
60 Frontage Road
Andover, MA 01810-5414
(617) 475-9090

Panel VI: Human Resources in PDSS

Government

Co-Chairman

Doldt, Ms. Linda
HQ AFLC/DPCS
Wright Patterson AFB, OH 45433-5000
(513) 257-4136
A/V 787-4136

Brooks, Ms. Sharon L.
HQ AD/ENE
Eglin AFB, FL 32542-5000
(904) 882-8505
A/V 872-8505

Cook, Ms. Lucille
Code 4023
Pacific Missile Test Center
Point Mugu, CA 93042
(805) 989-9405
A/V 351-9405

Fowler, Ms. Priscilla
Software Engineering Institute
Carnegie-Mellon University
580 South Aiken
Pittsburgh, PA 15232
(412) 268-7748

Green, Mr. Dan
Code N305
NSWC
Dahlgren, VA 22448
(703) 663-4673
A/V 249-4673

Harris, Mr. Andrew D.
LMC, Materiel Division
Installations and Logistics Department
Headquarters, U.S. Marine Corps
Washington, DC 20380-0001
(202) 695-4788
A/V 225-4788

Ipsen, Mr. Karl E.
CECOM
AMSEL-RD-LC-AST
Fort Monmouth, NJ 07703-5000
(201) 532-5831
A/V 992-5831

Lipke, Mr. Walter H.
Software Support Section
OC-ALC/MADAS
Tinker AFB, OK 73145-5990
(405) 736-5066
A/V 336-5066

Oglesby, Mr. Charles E.
HQ AMC-AMCDE-SB
5001 Eisenhower Ave.
Alexandria, VA 22333
(703) 274-9314
A/V 284-9314

Przybylinski, Mr. Stan
Software Engineering Institute
Carnegie-Mellon University
580 South Aiken
Pittsburgh, PA 15232
(412) 268-6371

Santo-Donato, Mr. Arthur S.
CECOM
AMSEL-RD-LC-SPM
Fort Monmouth, NJ 07703-5000
(201) 544-4353
A/V 995-4353

Simmons, Capt Denise J.
MCTSSA
Camp Pendleton, CA 92055-5080
(619) 725-2585
A/V 365-2585

Panel VI: Human Resources in PDSS

Industry

Co-Chairman

Brim, Mr. Terry
TRW
1628 Springfield St.
Dayton, OH 45403
(513) 253-0465

Mendis, Mr. Ken
Raytheon Co.
Box 360
Portsmouth, RI 02871
(401) 847-8000

Panel VII: Software Technology Transition

Government

Co-Chairman

Holinko, Mr. Myron
CECOM
AMSEL-RD-LC-IEW2
Fort Monmouth, NJ 07703-5000
(201) 544-3472
A/V 995-3472

Bates, Mr. Wayne
OO-ALC/MMARC
Hill AFB, UT 84056
(801) 777-6711
A/V 458-6711

Bedar, Maj George R.
MCTSSA
Camp Pendleton, CA 92055
(619) 725-2585
A/V 365-2585

Calland, Mr. Robert
Code 624(B)
NOSC
San Diego, CA 92152-5122
(619) 225-6231
A/V 933-6231

Glushko, Mr. Robert
Software Engineering Institute
Carnegie-Mellon University
580 South Aiken
Pittsburgh, PA 15232
(412) 268-6377

Malinowski, Mr. Gregory J.
CECOM
AMSEL-RD-LC-IEW2
Fort Monmouth, NJ 07703
(201) 544-2288
A/V 995-2288

McDonald, Mr. James
AFWAL/AAAF-3
Wright Patterson AFB, OH 45433-6543
(513) 255-6548
A/V 785-6548

Rodriguez, Mr. Albert
CECOM
AMSEL-RD-LC-COM-3B
Fort Monmouth, NJ 07703
(201) 532-4725
A/V 995-4725

Wasilausky, Mr. Robert
NOSC, Code 423
San Diego, CA 92152
(619) 225-2083
A/V 933-2083

Panel VII: Software Technology Transition

Industry

Co-Chairman

Marciniak, Mr. John
Marciniak and Associates
P.O. Box 2383
Arlington, VA 22202
(703) 920-9116

Baker, Dr. Emanuel R.
Software Engineering Consultants
10219 Briarwood Drive
Los Angeles, CA 92110
(213) 278-7241

Bracker, Dr. Lynne C.
Hughes Aircraft Company
P.O. Box 11337
Tuscon, AZ 85734-1337
(602) 794-5415

Cooper, Mr. Jack
Anchor Software Management, Ltd
Suite 214
5109 Leesburg Pike
Falls Church, VA 22041
(703) 578-3200

Harvey, Mr. Lawrence
Teledyne Brown Engineering
788 Shrewsbury Ave.
Tinton Falls, NJ 07724
(201) 741-5008

Irwin, Mr. Allen T.
SAIC
3045 Technology Parkway
Orlando, FL 32826
(305) 282-6700

Murray, Mr. William M.
General Dynamics
Data Systems Division
12101 Woodcrest Executive Drive
St. Louis, MO 63141
(314) 851-8910

Nuhn, Mr. Perry
Software Productivity Consortium, Inc.
1880 N. Campus Commons Drive
Reston, VA 22091
(703) 391-1718

Preston, Dr. David G.
IIT Research Institute
Suite 300
4550 Forbes Boulevard
Lanham, MD 20706-4324
(301) 731-8894

Smith, Mr. Jerry
QSOFT
Suite 206
2755 Hartland Road
Falls Church, VA 22043
(703) 560-4440

Panel VIII: MCCR Security

Government

Co-Chairman

Muzik, Ms. Sharon
NESEA, Code 2241
Bldg. 141
St. Inigoes, MD 20684
(301) 862-8436
A/V 356-3512
Washington Tie Line 870-2600 ext 8436

Cogar, Capt Gerald T.
AFCSC/SR
San Antonio, TX 78243-5000
(512) 925-2955
A/V 945-2955

Cole, Mr. John
CECOM
AMSEL-RD-COMM-TC2
Fort Monmouth, NJ 07703
(201) 544-4094
A/V 995-4094

Imler, Mr. David
HQ AFSC/SIS
Andrews AFB, MD 20334-5000
(301) 981-6450
A/V 858-6450

Lubbes, Mr. H.O.
SPAWAR, Code 3213
CP-6, Room 944
Washington, DC 20363
(202) 692-3966
A/V 222-3966

Meyers, Capt Glenn
HQ AFSC/PLRT
Andrews AFB, MD 20334-5000
(301) 981-6941
A/V 858-6941

Panel VIII: MCCR Security

Industry

Co-Chairman

Converse, Mr. Bob
Computer Sciences Corporation
6565 Arlington Blvd.
Falls Church, VA 22046
(703) 876-1210

Danner, Ms. Bonnie
Logicon
Suite 601
1555 Wilson Blvd.
Arlington, VA 22209
(703) 525-2484

Feldman, Mr. Charles
JASAR, Inc.
305 Dellwood Court
Annapolis, MD 21401
(301) 266-0698

Weidner, Mr. Michael
Sytek, Inc.
1945 Charleston Rd.
Mountain View, CA 94043
(415) 960-3400

☆ U.S. GOVERNMENT PRINTING OFFICE: 1989-201-403/72212