

UNCLASSIFIED

AD-A197 038
Copy 18 of 189 copies

AD-A197 038

DTIC FILE

2

IDA PAPER P-2018

GUIDELINES FOR TAILORING DOD-STD-2167A
FOR SDS SOFTWARE DEVELOPMENT

Robert J. Knapper
Cathy Jo Linn
John Salasin

DTIC
ELECTE
AUG 17 1988
S D

February 1988

Prepared for
Strategic Defense Initiative Organization (SDIO)

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited



INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, Virginia 22311

88 8 17

UNCLASSIFIED

IDA Log No. HQ 87-32194

DEFINITIONS

IDA publishes the following documents to report the results of its work.

Reports

Reports are the most authoritative and most carefully considered products IDA publishes. They normally embody results of major projects which (a) have a direct bearing on decisions affecting major programs, or (b) address issues of significant concern to the Executive Branch, the Congress and/or the public, or (c) address issues that have significant economic implications. IDA Reports are reviewed by outside panels of experts to ensure their high quality and relevance to the problems studied, and they are released by the President of IDA.

Papers

Papers normally address relatively restricted technical or policy issues. They communicate the results of special analyses, interim reports or phases of a task, ad hoc or quick reaction work. Papers are reviewed to ensure that they meet standards similar to those expected of refereed papers in professional journals.

Memorandum Reports

IDA Memorandum Reports are used for the convenience of the sponsors or the analysts to record substantive work done in quick reaction studies and major interactive technical support activities; to make available preliminary and tentative results of analyses or of working group and panel activities; to forward information that is essentially unanalyzed and unevaluated; or to make a record of conferences, meetings, or briefings, or of data developed in the course of an investigation. Review of Memorandum Reports is suited to their content and intended use.

The results of IDA work are also conveyed by briefings and informal memoranda to sponsors and others designated by the sponsors, when appropriate.

The work reported in this document was conducted under contract MDA 903 84 C 9031 for the Department of Defense. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that agency.

This paper has been reviewed by IDA to assure that it meets high standards of thoroughness, objectivity, and sound analytical methodology and that the conclusions stem from the methodology.

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

AD-A197038

1a REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS		
2a SECURITY CLASSIFICATION AUTHORITY			3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release - distribution unlimited		
2b DECLASSIFICATION/DOWNGRADING SCHEDULE					
4 PERFORMING ORGANIZATION REPORT NUMBER(S) IDA Paper P-2018			5 MONITORING ORGANIZATION REPORT NUMBER(S)		
6a NAME OF PERFORMING ORGANIZATION Institute for Defense Analyses		6b OFFICE SYMBOL IDA	7a NAME OF MONITORING ORGANIZATION OUSDA-DIMO		
6c ADDRESS (City, State, and Zip Code) 1801 N. Beauregard St. Alexandria, VA 22311			7b ADDRESS (City, State, and Zip Code) 1801 N. Beauregard Street Alexandria, Virginia 22311		
8a NAME OF FUNDING/SPONSORING ORGANIZATION Strategic Defense Initiative Organization		8b OFFICE SYMBOL (if applicable) SDIO	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MDA 903 84 C 0031		
8c ADDRESS (City, State, and Zip Code) SDIO/SYS/BM Room 1E149 Pentagon, Washington D.C. 20301-7100			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO. T-R5-422
			WORK UNIT ACCESSION NO.		
11 TITLE (Include Security Classification) Guidlines for Tailoring DOD-STD-2167A for SDS Software Development (U)					
12 PERSONAL AUTHOR(S) Robert J. Knapper, Cathy Jo Linn, John Salasin					
13a TYPE OF REPORT Final		13b TIME COVERED FROM _____ TO _____		14 DATE OF REPORT (Year, Month, Day) 1988 February	
				15 PAGE COUNT 53	
16 SUPPLEMENTARY NOTATION					
17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	Software Engineering; Software Development; Military Standards and Specifications; Data Item Description (DID); Software; Prototype; DOD-STD-2167A; Request for Proposal (RFP); Tailoring; Software Design; Concept Formulation (CF); Initial Prototype (IP); Advanced Prototype (AP); Full-Scale Development (FSD); MIL-STD-1521B		
19 ABSTRACT (Continue on reverse if necessary and identify by block number)					
IDA Paper P-2018 provides the SDIO with guidelines for tailoring of DOD-STD-2167A and related Data Item Descriptions (DIDs), including a suggested interpretation of the review requirements in MIL-STD-1521B for use as guidelines in developing prototype and production software. Although not approved for use, revision A of DOD-STD-2167 was chosen as the subject since it will supercede the current standard in 1987, and because it more adequately addresses modern software technology ideas. The recommendations in this paper are intended to be reflected in Requests for Proposals (RFPs) distributed to vendors proposing to develop SDI software and in contracts let for SDI software aquisition.					
20 DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS			21 ABSTRACT SECURITY CLASSIFICATION Unclassified		
22a NAME OF RESPONSIBLE INDIVIDUAL Dr. Cathy Jo Linn, IDA			22b TELEPHONE (Include Area Code) (703) 824-5520		22c OFFICE SYMBOL IDA/CSFD

UNCLASSIFIED

IDA PAPER P-2018

GUIDELINES FOR TAILORING DOD-STD-2167A
FOR SDS SOFTWARE DEVELOPMENT

Robert J. Knapper
Cathy Jo Linn
John Salasin

February 1988

Accession For:	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



INSTITUTE FOR DEFENSE ANALYSES

Contract MDA 903 84 C 0031
Task T-R5-422



UNCLASSIFIED

UNCLASSIFIED

TABLE OF CONTENTS

1. PURPOSE	1
2. SCOPE	1
3. BACKGROUND	1
3.1 SDI Software Needs	1
3.2 DOD-STD-2167A	2
3.3 Related Efforts	4
4. RECOMMENDED TAILORINGS	4
4.1 Additional Definitions	4
4.2 General Requirements Tailoring	5
4.3 Detailed Requirements Tailoring	10
4.4 Data Item Descriptions	10
4.5 Documentation and Review Requirements (MIL-STD-1521B)	10
4.5.1 Concept Formulation	11
4.5.2 Initial Prototypes	12
4.5.3 Advanced Prototypes	13
4.5.4 Full-Scale Development	14
5. CONCLUSIONS	14
APPENDIX A. LIST OF ACRONYMS	15
APPENDIX B. Applicable Data Item Description Tailorings	17
B.1 Software Development Plan (DI-MCCR-80030A)	17
B.2 Software Requirements Specification (DI-MCCR-80025A)	18
B.3 Software Product Specification (DI-MCCR-80029A)	18
APPENDIX C. REFERENCES	19
APPENDIX D. DOD-STD-2167A (27 OCTOBER 1987 DRAFT)	21

UNCLASSIFIED

THIS PAGE INTENTIONALLY BLANK

UNCLASSIFIED

PREFACE

This paper presents guidelines for tailoring the DOD-STD-2167A Software Development Standard to software that may be developed for the Strategic Defense Initiative Organization (SDIO). It is based on a preliminary examination of how DOD-STD-2167A might apply to Strategic Defense System (SDS) software in various stages of research and development. It provides general guidelines for use by SDIO, agents, and contractors.

The document, developed under Task Order T-R5-422, is part of IDA's effort to support the SDIO by providing analyses and recommendations for software activities. The document was reviewed on 16 July 1987 by the members of the following CSED Peer Review: Audrey Hook, Jack Kramer, Catherine McDonald, Sarah Nash, and Robert Winner. In addition, the draft version was externally reviewed by and comments were received from members of the SDIO community.

UNCLASSIFIED

THIS PAGE INTENTIONALLY BLANK

UNCLASSIFIED

ACKNOWLEDGEMENTS

The authors would like to express their thanks to the CSED peer review members, Audrey Hook, Jack Kramer, Catherine McDonald, Sarah Nash, and Robert Winner for their time and assistance in providing helpful comments for the final paper. Additional comments and guidance in preparing the final paper were gratefully accepted from the CSED Editor, Katydean Price. CSED staff secretary Jayne Hoblin prepared the figures used in the paper.

Appreciation is also expressed for comments received from reviewers outside of IDA: Danny Cohen - ISI, Albert Small - SRS Technologies, Lt Col Ted Mervosh - ESD Hanscom, Major Frank Maressa - SDIO, James Tremlett - RADDC, and LT Marcia Van Wye - USSPACECOM.

UNCLASSIFIED

THIS PAGE INTENTIONALLY BLANK

X
UNCLASSIFIED

UNCLASSIFIED

Guidelines for Tailoring DOD-STD-2167A for SDI Related Software Development

1. PURPOSE

This paper provides the SDIO with a recommended tailoring of DOD-STD-2167A and related Data Item Descriptions (DIDs), including a suggested interpretation of the review requirements in MIL-STD-1521B (Technical Reviews and Audits for Systems, Equipments, and Computer Software), for use as guidelines in developing prototype and production software. Although still a draft, Revision A of the Standard was chosen as the subject of this paper since it will supercede the current standard in 1987 and because it more adequately addresses modern software technology ideas. This paper does not suggest that changes be made to the Standard, but rather should be viewed as an interpretation of the Standard. However, based in part on recommendations derived from the initial draft of this paper and other work at IDA, some changes in the Standard have been made. The guidelines in this paper are intended to be reflected in Requests for Proposals (RFPs) distributed to vendors proposing to develop SDI software and in contracts let for SDI software acquisition.

2. SCOPE

Section 3. of this paper outlines the software needs of the SDI and describes how those needs may not be fulfilled with an untailored DOD-STD-2167A. (For reference, a copy of the Standard is found in Appendix D.) The specific tailoring recommendations for DOD-STD-2167A are found in Section 4. and the applicability of MIL-STD-1521B reviews over the range of potential SDI software programs is discussed. Appendix B covers the recommendations for tailoring the associated DIDs. Some notes and disclaimers are in Section 5.

3. BACKGROUND

3.1 SDI Software Needs

The software requirements of SDI Battle Management/Command, Control, and Communications (BM/C3) stretch beyond the current state-of-the-practice in performance, capacity, and adaptability. To meet those requirements in the future, software design and implementation will need to be innovative and must evolve rapidly with the changing edge of technology. Ideas will have to be translated quickly into software prototypes to demonstrate the validity of the technology they represent. In addition, libraries of *certified*¹ software components produced/acquired for SDI or other Government funded programs will be heavily used.

All aspects of software, from early-on studies of theoretical concepts expressed formally as

¹Certified in this case refers to software components guaranteed to be free of 'Trojan Horses' or other hostile entities and which are functionally correct to their specifications.

UNCLASSIFIED

abstract equations, represented as finite state automata, etc., through to the development, testing, and deployment of entire systems, are actively being pursued today by the computing community for use in a wide range of applications. Although interested in recognizing and supporting emerging technologies, the SDIO may only fund software efforts that result in products expressed in executable software, which are applicable to and/or extensible into Strategic Defense System (SDS) subsystems. Ideally, but not exclusively, the software developed for one phase of the program will be upwardly compatible with succeeding phases. That is, the final software product from one phase becomes the initial specifications for a succeeding phase. This differs from a more traditional approach of funding programs that do not rely on or do not take advantage of prior products as an initial baseline. BM/C3 potential software programs will fall into four general phases:

1. Concept Formulation (CF) - where software is developed to assist in proving concepts, defining technology requirements, or for demonstrating the feasibility or appropriateness of technical approaches.
2. Initial Prototypes (IP) - where software is developed to provide an initial demonstration of components in a system context to assist in defining interfaces and in refining functional requirements.
3. Advanced Prototypes (AP) - where systematic, incrementally advanced, integratable software implementations of the initial prototypes provide detailed functional and performance requirements for system development.
4. Full-Scale Development (FSD) - where complete operational code, evaluated against specific full-scale engineering decision (FSED) criteria, is developed for real-world applications.

The SDIO classifies programs in the CF and IP categories as "technology" programs, while those in AP (and possibly a small number in IP) are classified as "experimental validation" programs. The important point in the classification scheme is not, however, the relationship of the categories to a funding classification, but rather the uses to which the results of a program are to be applied. As a consequence, the requirements and the applicability of military standards will vary among the four categories.

3.2 DOD-STD-2167A

The Defense System Software Development Standard, DOD-STD-2167A (27 October 1987 draft), contains the requirements for the development of mission-critical computer resources (MCCR) software. DOD-STD-2167A is intended to establish "uniform requirements for software development that are applicable throughout the system life cycle." Although the Standard is intended for use through the entire life cycle and "is not intended to specify or discourage the use of any particular software development method", it must be tailored to suit the specific contract needs of individual programs. In general, the Standard provides an adequate mechanism for the software development process. Some parts however, require a clarification, additional emphasis, or a specific interpretation to support the software philosophies held by the SDIO.

FSD software will make heavy use of reusable modules from non-developmental software (NDS) libraries of Government furnished software (GFS) components or certified Commercial Off-The Shelf (COTS) software. Procedures for developing, cataloging, storing, accessing, and

UNCLASSIFIED

controlling certified software components in reusable component libraries will be very complex. A hierarchy of attributes relating to the security certification, fault tolerance, and verifiability of each library entry will have to be established and maintained through an intelligent library management system to make effective reuse of software. A straight-forward top-down design approach would not naturally lend itself to constructing software from these reusable modules. The reuse of software modules often requires an element of the bottom-up approach to design. The Standard supports reuse by requiring the contractor to consider using NDS software (para 4.2.4). However, the specific requirement to exploit reusable software is not emphasized. Tailoring is needed to require development methodologies that apply or encourage this practice.

Similarly, using the classic *waterfall* life cycle methodology within research activities is constraining. The waterfall method is an unambiguous method for implementing production software, beginning with requirements and proceeding linearly through specifications, design, implementation, and maintenance. With this method however, errors in requirements, specification, and design are not detected until the software is implemented. Finding errors so late in the life cycle adversely impacts not only cost but also delivery schedules. Errors in early phases are especially common in development projects for systems that have not been previously implemented and for which the requirements are not well-defined or understood. To avoid this pitfall, the SDIO is employing a *prototyping* discipline whereby key portions of the system are developed to ensure the correctness of the system requirements and specifications. Errors detected in the prototyping efforts result in early corrections to the system specification, saving time and helping to keep costs within budget. The Standard supports the use of prototyping by specifically requiring that the "software development process shall include the following major activities, which may overlap and may be applied iteratively or recursively..." (para 4.1.1). Also, the contractor is required to use a systematic and well-documented software development method (para 4.2.1). Prototyping is such a method. However, there is no explicit requirement for prototyping. The SDIO promotion of a prototyping development methodology needs to be reflected in the tailorings.

The SDI software will be developed by a variety of organizations ranging from universities to aerospace firms. The application of DOD-STD-2167A's documentation requirements (para 4.6.4 and Figure 2.) needs to be tailored appropriately. For example, the impact of a CF activity conducting technology feasibility studies might become reduced if the activity is mired in formal documentation. It is reasonable that a minimum of formal documentation may be all that is needed with an IP prototype. On the other hand, an FSD activity may not only be constructing an operational product, as opposed to an experimental prototype, but would have well-defined requirements and specifications to use as the required formal documentation. Along the same lines, the formal reviews and audits as specified in MIL-STD-1521B are less applicable for CF and IP activities. The Standard supports this view by allowing the degree to which formal reviews are required to be determined by the contract (para 4.1.2). This is relatively explicit and should not need further clarification through tailoring. However, some suggested documentation and review requirements for various activities are shown in Section 4.4.

Finally, the Standard does not explicitly require the use of Ada² and Ada program design language (PDL) as stated in DoD Directives 3405.1 and 3405.2. Even if a waiver for use of a language other than Ada will be requested, the Standard needs to be tailored to reflect

²Ada is a registered trademark of the U.S. Government, Ada Joint Program Office

UNCLASSIFIED

conformity with DoDDs 3405.1 and 3405.2. In addition, the use of SDI Ada/Process Description Language (SA/PDL) to describe the design of information processing elements of the SDS will be required.

3.3 Related Efforts

In addition to applying military standards to the software development process as a whole, other efforts, such as for testing and evaluation, are underway to "standardize" specific aspects of software development.

A Test and Evaluation Master Plan (TEMP) is being developed to provide guidance for the planning, execution, and reporting of BM/C3 software test and evaluation (T&E) activities. This TEMP will differ from other TEMPs for large and complex software systems, such as the World Wide Military Command and Control Information System (WIS), in that this TEMP identifies the need for formal notations to support the rigorous specification of testing requirements as part of software requirements and specifications.

Since any system cannot be tested except against its specifications, T&E will be facilitated by using a formal notation that provides a grammar for ensuring complete specification of critical software characteristics, their threshold values, and associated testing requirements.

4. RECOMMENDED TAILORINGS

Tailoring is a cost-effective method for applying the requirements to meet the specific needs of particular programs. A well-tailored standard will avoid unnecessary activities during the acquisition and will thereby reduce the cost of the overall software development effort.

The tailoring of the Standard involves three general steps: tailoring the General Requirements in Section 4., tailoring the Specific Requirements in Section 5., and finally tailoring the associated DIDs. To facilitate an understanding of some of the tailorings, additional definitions have been suggested for Section 3. All paragraph or figure references that follow are from the Standard or DIDs. All changes within a paragraph will be denoted with change bars and highlighted with boldface type.

4.1 Additional Definitions

Additions to Section 3. DEFINITIONS.

3.XX Development phases:

3.XX.1 Concept Formulation. Development of software to assist in proving concepts, defining technology requirements, or demonstrating the feasibility or appropriateness of technical approaches.

3.XX.2 Initial Prototypes. Development of software to provide an initial demonstration of components in a system context to assist in defining interfaces and in refining functional requirements.

3.XX.3 Advanced Prototypes. Development of systematic, incrementally advanced, integratable software implementations of initial prototypes to provide detailed functional and performance requirements for system development.

3.XX.4 Full-Scale Development. Development of operational software systems,

UNCLASSIFIED

undertaken based on an evaluation of prototype and study results with respect to specific full-scale engineering decision criteria.

4.2 General Requirements Tailoring

Changes in Section 4. GENERAL REQUIREMENTS.

para 4.1.1 Software development process. ... The software development process shall include the following major activities, as applicable to the development phase (see Figures 2a-2d) and as specified in the contract:

- I a. Preliminary Software Requirements Analysis
- I b. Preliminary Design (Initial Prototype)
- I c. Iterative Prototype Design Change/Refinement/Expansion
- I d. Refined Software Requirements Analysis
 - e. Detailed Design
 - f. Coding and Unit Testing
 - g. CSC Integration and Testing
 - h. CSCI Testing

The last item "i", is deleted.

I para 4.1.2 Formal reviews/audits. ... Figures 2a through 2d illustrate the occurrence of formal reviews and audits for the four software development phases and shows the relationship of the minimal set of deliverable products to baselines and the Developmental Configuration.

para 4.1.9 Software development library. ... The contractor shall document and implement procedures for controlling software and associated documentation residing within the SDL. The contractor shall also document and implement procedures for cataloging, storing, accessing, and controlling reusable software. The contractor shall maintain the SDL ...

I para 4.2.4 Non-developmental software. The contractor shall, to the maximum extent possible, incorporate non-developmental software (NDS), such as reusable software that was internally developed or obtainable as Government furnished software (GFS), into the deliverable software. The contractor shall document plans for using NDS. NDS shall be catalogued, stored, and controlled through the procedures for the software development library. NDS may be ...

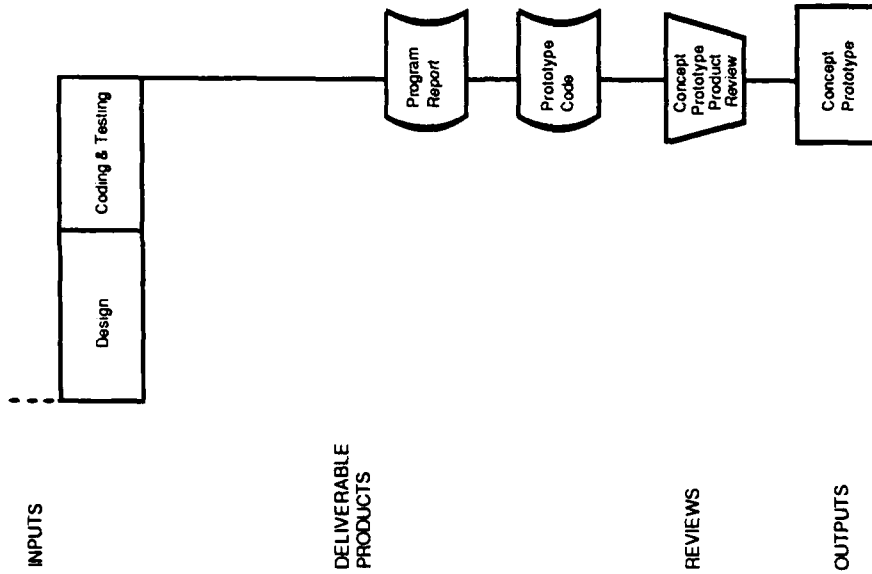


FIGURE 2a. Concept formulation activities and products.

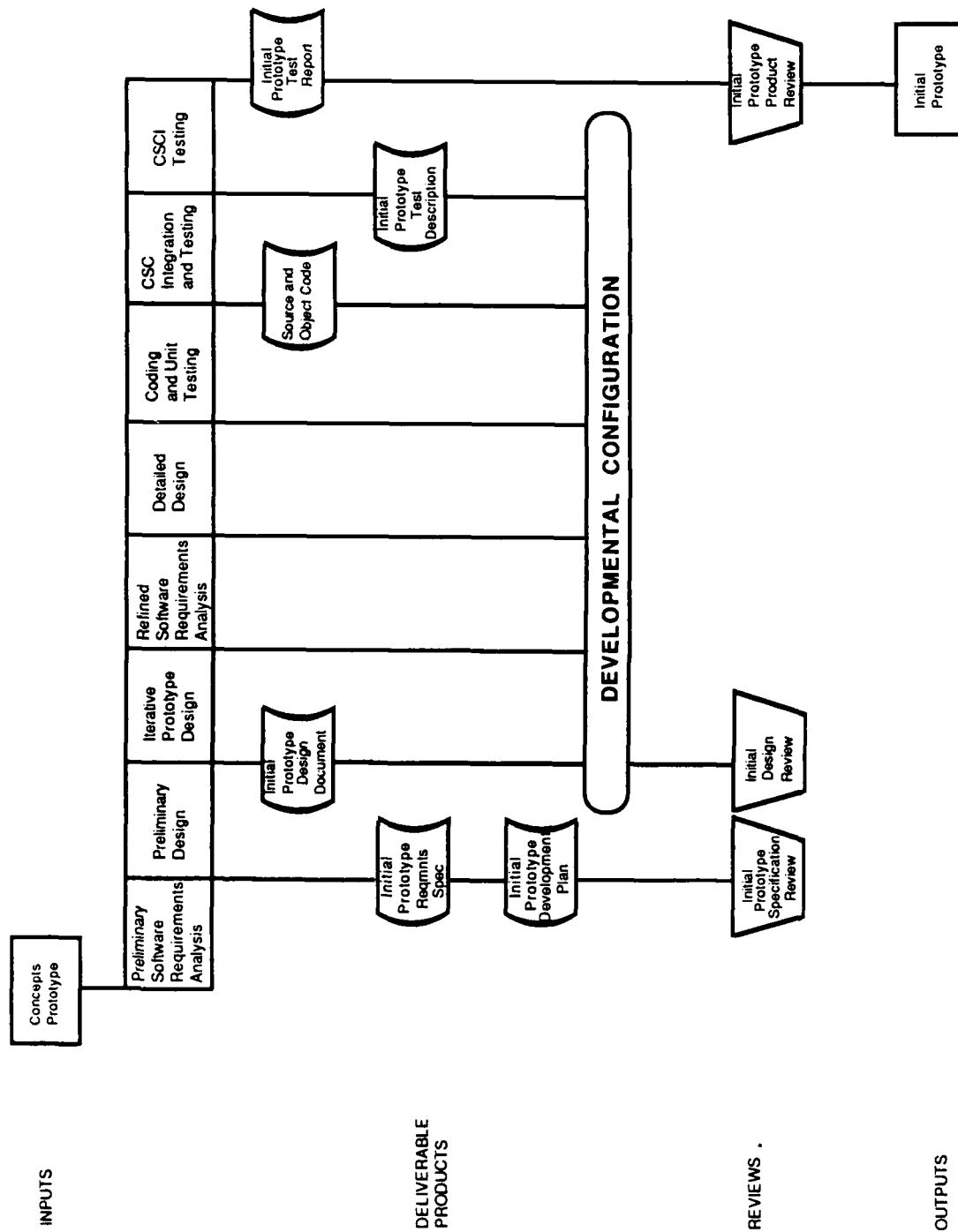


FIGURE 2b. Initial prototypes activities and products.

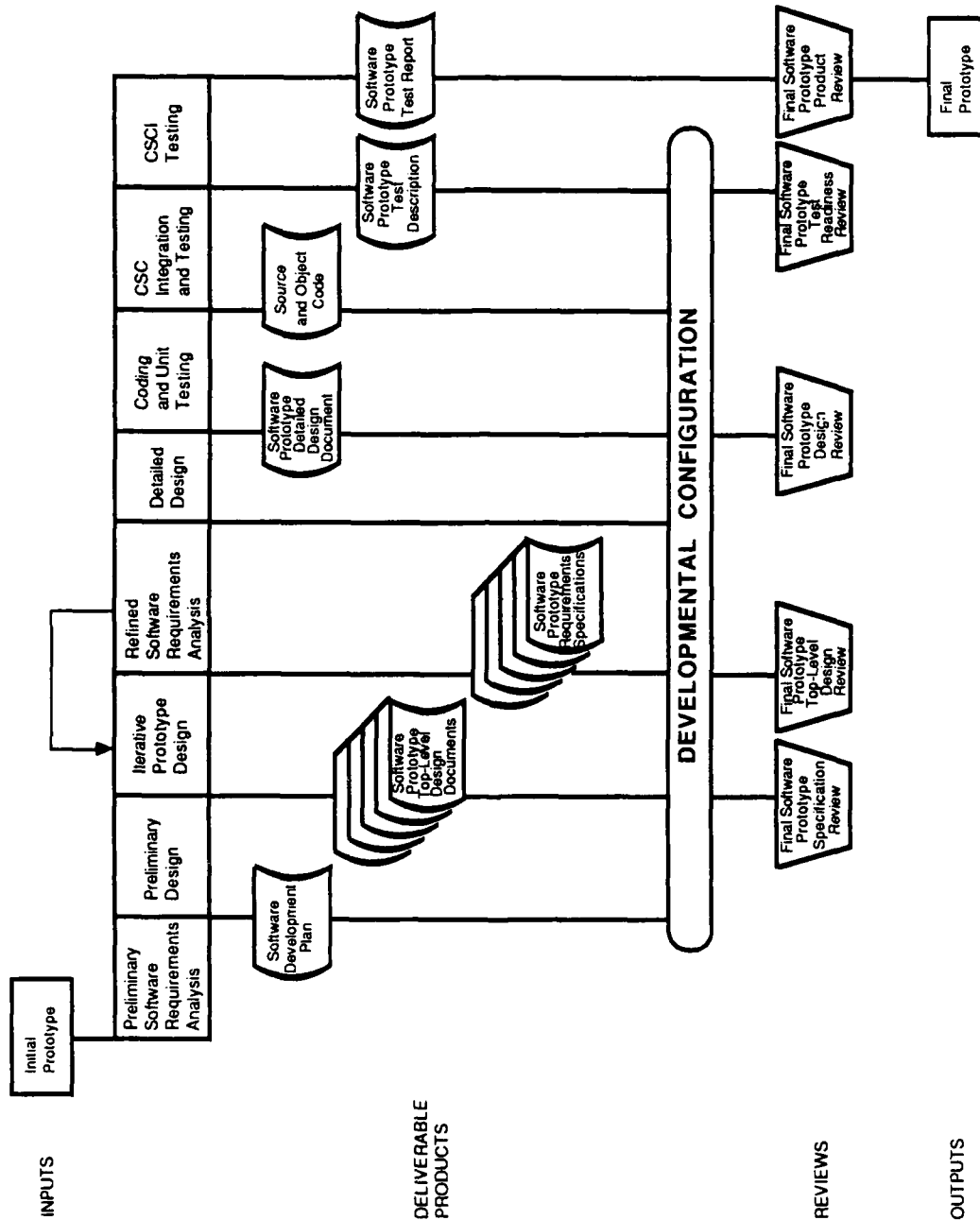


FIGURE 2c. Advanced prototypes activities and products.

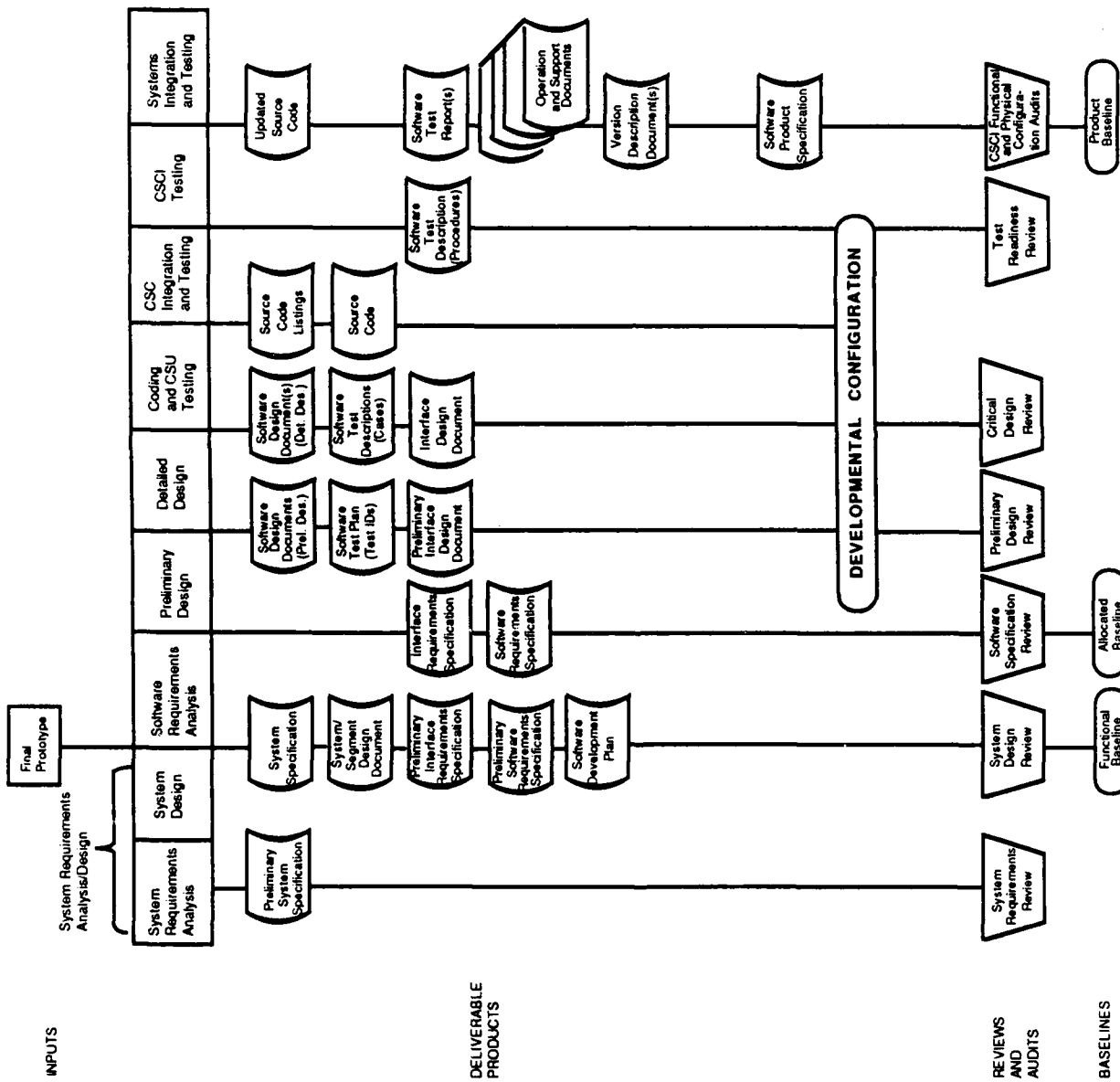


FIGURE 2d. Full-scale development activities and products.

UNCLASSIFIED

para 4.2.7 High order language. The contractor shall use the High Order Language (HOL) Ada, as required by DoD Directives 3405.1 and 3405.2, to code deliverable software.

The last sentence in para 4.2.7, "If no HOL ...", is deleted.

para 4.2.8 Design and coding standards. The contractor shall document and implement the Ada PDL and Ada coding standards to be used in the development of deliverable software. In addition, the contractor shall use SDI Ada Process Description Language to provide a formal description of the design for simulation and evaluation prior to the development of code. Software coding standards shall comply with the requirements specified in Appendix B.

4.3 Detailed Requirements Tailoring

Changes in Section 5. DETAILED REQUIREMENTS.

No changes.

4.4 Data Item Descriptions

The DIDs associated with DOD-STD-2167A describe a concise and complete set of documents for recording information required by the Standard. As with the Standard itself, the DIDs may be tailored. The Contract Data Requirements List (CDRL), DD Form 1423, incorporated into a contract identifies the data requirements that shall be developed as specified by approved DIDs. Not every DID associated with the Standard is applicable to every contract however. Appendix B in this paper provides a tailoring of those DIDs requiring modification to maintain consistency with the Standard as tailored by this paper. Applicability, tailoring, etc. of the other DIDs is contract dependent. In addition, there are items for which DIDs do not exist, e.g. the Program Report for CF activities. Preparation of draft DIDs for these items may be considered as a future effort.

4.5 Documentation and Review Requirements (MIL-STD-1521B)

The SDIO will potentially fund programs at all four phases of development. As stated before, these programs are varying in their scope of activities and deliverable products. Therefore, it is reasonable that the documentation and review requirements across these four phases will also vary. Fundamental research performing exploratory development of experimental prototypes may not reach the stage of detailed design, while full-scale engineering development programs will develop completely integrated, operationally tested products.

Specific requirements for documentation and reviews are individually contract dependent and the guidelines for tailoring MIL-STD-1521B are found in Appendix J of that standard. The Standard is very clear in paragraph 100.4.1, "The key to tailoring MIL-STD-1521 is to match the MIL-STD-1521 requirements against the details of the applicable SOW/Contractual task requirements." The following though, may serve as a possible generic solution or as a baseline

UNCLASSIFIED

for BM/C3 software programs. The mapping of the documentation and reviews listed for the IP and AP phases into the DIDs and MIL-STD-1521B is shown in parentheses.

4.5.1 Concept Formulation

The requirements for CF phase programs are relatively simple. Developing software to prove concepts or for defining requirements is not aimed at producing an operational implementation but rather at developing technology into a feasible demonstration with minimal risks. None of the documents required by DOD-STD-2167A nor the reviews required by MIL-STD-1521B are applicable. However, all SDIO funded activities are expected to generate products, either in the form of documents, code, or both. For CF programs the following will be required:

Documentation:

- * Program Report consisting of textual documentation describing:
 - Capabilities and functionality of the technology developed
 - Interface with other SDI activities
 - Benefits provided to SDI
 - Approach used to develop this technology
 - Test results from simulating the prototype
- * Concepts Prototype Code, either as Ada, SA/PDL, or a specifically approved language

Reviews:

- * Concepts Prototype Product Review

4.5.2 Initial Prototypes

IP programs will develop the initial prototypes in SA/PDL, often using the technology demonstrated in CF programs. Preliminary requirements will be derived directly from CF code. Test plans and procedures will also be developed; testing will be conducted and reported. An IP program would need the following documents and formal reviews:

Documentation:

- * Initial Prototype Requirements Specification (*Preliminary Software Requirements*)

UNCLASSIFIED

Specification)³

- * Initial Prototype Development Plan (*Preliminary* Software Development Plan)
- * Initial Prototype Design Document (*Preliminary* Software Top-Level Design Document)
- * Initial Prototype Source and Object code
- * Initial Prototype Test Description (*Preliminary* Software Test Description)
- * Initial Prototype Test Report (*Preliminary* Software Test Report)

Reviews:

- * Initial Prototype Specification Review (*Preliminary* Software Specification Review)
- * Initial Prototype Design Review (*Preliminary* Design Review)
- * Initial Prototype Product Review

4.5.3 Advanced Prototypes

Programs in the AP phase will involve systematic, incremental growth of IP prototypes and will result in very nearly the implementation of a complete system or integratable component capable of handling real-world operational constraints in the simulation system. The system design will be taken directly from the SA/PDL of the preceding IP. Structural, functional, and interface organization and procedures will be documented in a software development plan. As the incremental prototypes are produced, tested, and refined, the final requirements and the final design documents will be produced. Test plans and procedures will also be produced; testing will be conducted and reported. An AP program should include the following documentation and reviews:

Documentation:

- * Software Prototype Development Plan (*Refined* Software Development Plan)
- * Software Prototype Requirements Specifications (*Refined* Software Requirements Specification)
- * Software Prototype Top-Level Design Documents (*Refined* Software Top-Level Design Document)
- * Software Prototype Detailed Design Document (Software Detailed Design Document)

³Items in "()" are the relevant DOD-STD-2167A DIDs

UNCLASSIFIED

- * Software Prototype Source and Object code
- * Software Prototype Test Description (*Refined* Software Test Description)
- * Software Prototype Test Report (*Refined* Software Test Report)

Reviews:

- * Final Software Prototype Specification Review (*Refined* Software Specification Review)
- * Final Software Prototype Top-Level Design Review (*Refined* Preliminary Design Review)
- * Final Software Prototype Design Review (Critical Design Review)
- * Final Software Prototype Test Readiness Review (Test Readiness Review)
- * Final Software Prototype Product Review

4.5.4 Full-Scale Development

A full-scale engineering development program will result in a completely operational, fully integrated, and thoroughly tested software system. The software requirements and detailed design will have been refined from AP programs. From this point, depending on the contract, it is reasonable to produce all of the documentation and prepare for all of the formal reviews specified in DOD-STD-2167A and MIL-STD-1521B by using the waterfall life cycle method as a model.

5. CONCLUSIONS

The 27 October 1987 draft of DOD-STD-2167A and 01 May 1987 drafts of the associated DIDs were used in this paper. This version of DOD-STD-2167A has been forwarded by the Joint Logistical Commanders Joint Policy Coordinating Group on Resource Management to the Defense Product Standards Office for release. Any changes made to the Standard prior to this release should not affect this paper.

As mentioned in the previous section, the requirements for documentation and reviews are contract specific. The SDIO and/or the contracting agency should work with the contractor to correctly specify the appropriate documents and reviews needed.

The Standard, although tuned to more modern methods of software development, must still be tailored to provide a structure for cost-effective software acquisition. The recommendations in this paper are untried. However, a good example, along the same lines, of a completely tailored Standard is currently in draft form out of the System Integration Office of the U.S. Space Command.

UNCLASSIFIED

THIS PAGE INTENTIONALLY BLANK

UNCLASSIFIED

APPENDIX A LIST OF ACRONYMS

ABM	Anti-Ballistic Missile
AP	Advanced Prototypes
BM/C3	Battle Management/Command, Control, Communications
CDRL	Contract Data Requirements List
CF	Concept Formulation
COTS	Commercial Off-the-Shelf
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
DID	Data Item Description
DOD	Department of Defense
FSD	Full-Scale Development
FSED	Full-Scale Engineering Decision
GFS	Government Furnished Software
HOL	High Order Language
IP	Initial Prototypes
LLCSC	Lower-Level Computer Software Component
MCCR	Mission-Critical Computer Resources
NDS	Non-Developmental Software
PDL	Program Design Language
RFP	Request for Proposals
SA/PDL	SDI Ada/Process Description Language
SDI	Strategic Defense Initiative
SDIO	Strategic Defense Initiative Organization
SDL	Software Development Library
SDP	Software Development Plan
SDS	Strategic Defense System
SOW	Statement of Work
TLCSC	Top-Level Computer Software Component
WIS	World Wide Military Command and Control Information System

UNCLASSIFIED

THIS PAGE INTENTIONALLY BLANK

APPENDIX B

Applicable Data Item Description Tailorings

B.1 Software Development Plan (DI-MCCR-80030A)

Changes in Section 10. PREPARATION INSTRUCTIONS

para 10.2.5.9 **Software development library.** This paragraph shall be numbered 3.9 and shall describe the software development library (SDL) to be used by the contractor for controlling the software and associated documentation. This paragraph shall include a description of the contractor's procedures and methods for establishing and implementing the SDL, the contractor's access and control procedures for data stored in the SDL, and the contractor's procedures for cataloging, storing, and accessing reusable component software.

para 10.2.6.2.1 **Software development techniques and methodologies.** This subparagraph shall be numbered 4.2.1 and identify and describe the techniques and methodologies the contractor plans to use to perform:

- I a. **Preliminary Software Requirements Analysis** (including structured requirements analysis tools, techniques, or a combination of both)
 - I b. **Preliminary Design (Initial Prototype)**
 - I c. **Iterative Prototype Design Change/Refinement/Expansion**
 - I d. **Refined Software Requirements Analysis**
 - e. Detailed Design
 - f. Coding and Unit Testing
 - g. CSC Integration and Testing
 - h. CSCI Testing
-

para 10.2.6.2.4 **Design standards.** This subparagraph shall be numbered 4.2.4 and shall reference the Ada PDL design standards the contractor will use in developing the software. This subparagraph shall also reference the SA/PDL used for process description.

The last sentence in para 10.2.6.2.4, "If the contractor ..." is deleted.

para 10.2.6.2.5 **Coding standards.** This subparagraph shall be numbered 4.2.5 and shall reference the Ada coding standards in the contractor will use in developing the software. Any planned modifications to the referenced coding standards shall be documented in this subparagraph.

The second sentence in para 10.2.6.2.5, "If the contractor ..." is deleted.

B.2 Software Requirements Specification (DI-MCCR-80025A)

para 10.2.5.3.1 **Programming language(s).** This subparagraph shall be numbered 3.3.1 and shall reference the Ada programming language required by DoDDs 3405.1 and 3405.2. to be used to implement the CSCI.

para 10.2.5.3.2 **Coding standards.** This subparagraph shall be numbered 3.3.2 and shall specify directly or by reference the Ada coding standards under which the CSCI shall be implemented.

para 10.2.5.3.3 **Compiler/assembler.** This subparagraph shall be numbered 3.3.3 and shall specify the Ada compiler to be used to translate the CSCI implementation.

para 10.2.5.4.1 **Design standards.** This subparagraph shall be numbered 3.4.1 and shall specify directly or by reference the Ada PDL standards under which the CSCI shall be implemented.

B.3 Software Product Specification (DI-MCCR-80029A)

para **Compiler/assembler.** This paragraph shall be numbered 3.5 and shall specify the Ada compiler used to translate the source code.

UNCLASSIFIED

APPENDIX C REFERENCES

1. Defense System Software Development Standard, DOD-STD-2167A, prepared by SPAWAR, 27 October 1987 draft.
2. *Software Development Plan*, DI-MCCR-80030A, Data Item Description, prepared by NAVY-EC, 01 May 1987 draft.
3. *Software Requirements Specification*, DI-MCCR-80025A, Data Item Description, prepared by NAVY-EC, 01 May 1987 draft.
4. *Software Product Specification*, DI-MCCR-80029A, Data Item Description, prepared by NAVY-EC, 01 May 1987 draft.
5. Technical Reviews and Audits for Systems, Equipments, and Computer Software, MIL-STD-1521B, 04 June 1985
6. *Computer Programming Language Policy*, Department of Defense Directive 3405.1, Under Secretary of Defense (Acquisition), 02 April 1987.
7. *Use of Ada in Weapon Systems*, Department of Defense Directive 3405.2, Under Secretary of Defense (Acquisition), 30 March 1987.
8. *Strategic Defense Initiative Ada Process Description Language*, Linn, C.J., et al, IDA Paper P-1983, 16 March 1987.
9. *Strategic Defense Initiative Software Test and Evaluation Master Plan (TEMP)*, Version 1.01 Draft, 20 May 1987.

UNCLASSIFIED

THIS PAGE INTENTIONALLY BLANK

UNCLASSIFIED

APPENDIX D
DOD-STD-2167A (27 OCTOBER 1987 DRAFT)

UNCLASSIFIED

THIS PAGE INTENTIONALLY BLANK

DOD-STD-2167A
27 October 1987

SUPERSEDING
DOD-STD-2167
4 June 1985

DRAFT

MILITARY STANDARD

DEFENSE SYSTEM

SOFTWARE DEVELOPMENT

AMSC NO. _____

AREA MCCR

DISTRIBUTION STATEMENT A.

Approved for public release:
distribution is unlimited.

DOD-STD-2167A (DRAFT)

DEPARTMENT OF DEFENSE
Washington, D.C. 20301

Defense System Software Development

1. This Military Standard is approved 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, Space and Naval Warfare Systems Command, ATTN: SPAWAR - 3212, Washington, D.C. 20363-5100, by using the self-addressed Standardization Document Improvement Proposal (DD Form 1426) appearing at the end of this document or by letter.

FOREWORD

1. This standard establishes uniform requirements for software development that are applicable throughout the system life cycle. The requirements of this standard provide the basis for Government insight into a contractor's software development, testing, and evaluation efforts. This standard is to be used in conjunction with DOD-STD-2168, Defense System Software Quality Program.
2. This standard is not intended to specify or discourage the use of any particular software development method. The contractor is responsible for selecting software development methods that best support the achievement of contract requirements.
3. This standard, together with the other DOD and military documents referenced in Section 2, provides the means for establishing, evaluating, and maintaining quality in software and associated documentation. Guidance on the implementation of this standard and its relationship to these other documents is provided in DOD-HDBK-287.
4. Data Item Descriptions (DIDs) applicable to this standard are listed in Section 6. These DIDs describe a set of documents for recording the information required by this standard. Production of deliverable data using automated techniques is encouraged.

CONTENTS

<u>Paragraph</u>	<u>Page</u>
1. SCOPE	1
1.1 Purpose	1
1.2 Application	1
1.2.1 System development	1
1.2.2 Software quality program	1
1.2.3 Firmware	1
1.2.4 Software developed by Government agencies	1
1.3 Tailoring of this standard	1
2. REFERENCED DOCUMENTS	3
2.1 Government documents	3
2.1.1 Specifications, standards, and handbooks	3
2.1.2 Other Government documents, drawings, and publications	3
2.2 Other publications	3
2.3 Order of precedence	3
3. DEFINITIONS	5
4. GENERAL REQUIREMENTS	9
4.1 Software development management	9
4.1.1 Software development process	9
4.1.2 Formal reviews/audits	9
4.1.3 Software development planning	9
4.1.4 Risk management	9
4.1.5 Security	9
4.1.6 Subcontractor management	9
4.1.7 Interface with the software IV&V agent	9
4.1.8 Cost/schedule reporting	9
4.1.9 Software development library	11
4.1.10 Corrective action process	11
4.1.11 Problem/change report	11
4.2 Software engineering	11
4.2.1 Software development methods	11
4.2.2 Software engineering environment	11
4.2.3 Safety analysis	11
4.2.4 Non-developmental software	14
4.2.5 Computer software organization	14
4.2.6 Traceability of requirements to design	14
4.2.7 High order language	14
4.2.8 Design and coding standards	14
4.2.9 Software development files	14
4.2.10 Processing resource and reserve capacity	14
4.3 Formal qualification testing	16
4.3.1 Formal qualification test planning	16
4.3.2 Software test environment	16
4.3.3 Independence in FQT activities	16
4.3.4 Traceability of requirements to test cases	16

CONTENTS - Continued

Paragraph	Page
4.4 Software product evaluations	16
4.4.1 Independence in product evaluation activities	16
4.4.2 Final evaluations	16
4.4.3 Software evaluation records	16
4.4.4 Evaluation criteria	17
4.5 Software configuration management	17
4.5.1 Configuration identification	17
4.5.2 Configuration control	17
4.5.3 Configuration status accounting	17
4.5.4 Storage, handling, and delivery of project media	18
4.5.5 Engineering Change Proposals	18
4.6 Transitioning to software support	18
4.6.1 Regenerable and maintainable code	18
4.6.2 Transition planning	18
4.6.3 Software transition and continuing support	18
4.6.4 Software support and operational documentation	18
5. DETAILED REQUIREMENTS	19
5.1 System requirements analysis/design	19
5.1.1 Software development management	19
5.1.2 Software engineering	19
5.1.3 Formal qualification testing	19
5.1.4 Software product evaluations	19
5.1.5 Configuration management	19
5.2 Software requirements analysis	21
5.2.1 Software development management	21
5.2.2 Software engineering	21
5.2.3 Formal qualification testing	21
5.2.4 Software product evaluations	21
5.2.5 Configuration management	21
5.3 Preliminary design	23
5.3.1 Software development management	23
5.3.2 Software engineering	23
5.3.3 Formal qualification testing	23
5.3.4 Software product evaluations	23
5.3.5 Configuration management	23
5.4 Detailed design	25
5.4.1 Software development management	25
5.4.2 Software engineering	25
5.4.3 Formal qualification testing	25
5.4.4 Software product evaluations	25
5.4.5 Configuration management	25
5.5 Coding and CSU testing	28
5.5.1 Software development management	28
5.5.2 Software engineering	28
5.5.3 Formal qualification testing	28
5.5.4 Software product evaluations	28
5.5.5 Configuration management	28

CONTENTS - Continued

<u>Paragraph</u>	<u>Page</u>
5.6 CSC integration and testing	30
5.6.1 Software development management	30
5.6.2 Software engineering	30
5.6.3 Formal qualification testing	30
5.6.4 Software product evaluations	30
5.6.5 Configuration management	30
5.7 CSCI testing	32
5.7.1 Software development management	32
5.7.2 Software engineering	32
5.7.3 Formal qualification testing	32
5.7.4 Software product evaluations	32
5.7.5 Configuration management	32
5.8 System integration and testing	34
5.8.1 Software development management	34
5.8.2 Software engineering	34
5.8.3 Formal qualification testing	34
5.8.4 Software product evaluations	34
5.8.5 Configuration management	34
6. NOTES	35
6.1 Data requirements list and cross reference	35
6.2 Subject term (key word) listing	36
INDEX	47

CONTENTS - Continued

FIGURES

<u>Figure</u>		<u>Page</u>
1	An example of system development reviews and audits	10
2	Deliverable products, reviews, audits, and baselines	12
3	Example of a system breakdown and CSCI organizational structure	15
4	Evaluation criteria for products of system requirements analysis/design	20
5	Evaluation criteria for products of Software Requirements Analysis	22
6	Evaluation criteria for products of Preliminary Design	24
7	Evaluation criteria for products of Detailed Design	27
8	Evaluation criteria for products of Coding and CSU Testing	29
9	Evaluation criteria for products of CSC Integration and Testing	31
10	Evaluation criteria for products of CSCI Testing	33

APPENDIXES

Appendix

A	List of acronyms and abbreviations	39
B	Requirements for software coding standards	41
C	Category and priority classifications for software problem reporting	43
D	Evaluation Criteria	45

1. SCOPE

1.1 Purpose. The purpose of this standard is to establish requirements to be applied during the acquisition, development, and support of software systems.

1.2 Application. This standard applies to the extent specified in the contract clauses, the Statement of Work (SOW), and the Contract Data Requirements List (CDRL).

1.2.1 System development. Application of this standard must be coordinated with MIL-STD-499, Engineering Management, for total system development.

1.2.2 Software quality program. Application of this standard must be coordinated with DOD-STD-2168, Defense System Software Quality Program, to provide a complete software quality program.

1.2.3 Firmware. This standard applies to the development or support of the software element of firmware. This standard does not apply to the development of the hardware element of firmware.

1.2.4 Software developed by Government agencies. The provisions of this standard may be applied to Government agencies. When a Government agency performs software development or support in accordance with this standard, the term "contractor" refers to that Government agency and the term "subcontractor" refers to any contractor of that Government agency.

1.3 Tailoring of this standard. This standard contains a set of requirements designed to be tailored for each contract by the contracting agency. The tailoring process intended for this standard is the deletion of non-applicable requirements.

2. REFERENCED DOCUMENTS

2.1 Government documents.

2.1.1 Specifications, standards, and handbooks. Unless otherwise specified, the following specifications, standards, and handbooks of the issue listed in that issue of the Department of Defense Index of Specifications and Standards (DODISS) specified in the solicitation form a part of this standard to the extent specified herein.

MILITARY STANDARDS

- DOD-STD-2168 - Defense System Software Quality Program
- DOD-STD-480 - Configuration Control - Engineering Changes, Deviations, and Waivers
- MIL-STD-481 - Configuration Control - Engineering Changes, Deviations, and Waivers (Short Form)
- MIL-STD-483 - Configuration Management Practices for Systems, Equipment, Munitions, and Computer Software
- MIL-STD-490 - Specification Practices
- MIL-STD-499 - Engineering Management
- MIL-STD-1521 - Technical Reviews and Audits for Systems, Equipments, and Computer Software

2.1.2 Other Government documents, drawings, and publications. None.

(Copies of specifications, standards, handbooks, drawings, and publications required by contractors in connection with specific acquisition functions should be obtained from the contracting agency or as directed by the contracting officer.)

2.2 Other publications. None.

2.3 Order of precedence. In the event of a conflict between the text of this standard and the references cited herein, the text of this standard shall take precedence.

3. DEFINITIONS

3.1 Allocated Baseline. See DOD-STD-480.

3.2 Authentication. Determination by the Government that specification content is acceptable.

3.3 Baseline. See DOD-STD-480.

3.4 Computer data definition. A statement of the characteristics of the basic elements of information operated upon by hardware in responding to computer instructions. These characteristics may include, but are not limited to, type, range, structure, and value.

3.5 Computer hardware. Devices capable of accepting and storing computer data, executing a systematic sequence of operations on computer data, or producing control outputs. Such devices can perform substantial interpretation, computation, communication, control, or other logical functions.

3.6 Computer resources. The totality of computer hardware, software, personnel, documentation, supplies, and services applied to a given effort.

3.7 Computer software (or software). A combination of associated computer instructions and computer data definitions required to enable the computer hardware to perform computational or control functions.

3.8 Computer Software Component (CSC). A distinct part of a computer software configuration item (CSCI). CSCs may be further decomposed into other CSCs and Computer Software Units (CSUs).

3.9 Computer Software Configuration Item (CSCI). A configuration item for computer software.

3.10 Computer software documentation. Technical data or information, including computer listings and printouts, which documents the requirements, design, or details of computer software, explains the capabilities and limitations of the software, or provides operating instructions for using or supporting computer software during the software's operational life.

3.11 Computer Software Unit (CSU). An element specified in the design of a Computer Software Component (CSC) that is separately testable.

3.12 Configuration Identification. See DOD-STD-480.

3.13 Configuration Item. See DOD-STD-480.

3.14 Contracting agency. As used in this standard, contracting agency refers to the "contracting office" as defined in Federal Acquisition Regulation Subpart 2.1, or its designated representative.

3.15 Developmental Configuration. The contractor's software and associated technical documentation that defines the evolving configuration of a CSCI during development. It is under the development contractor's configuration control and describes the software design and implementation. The Developmental Configuration for a CSCI consists of a Software Design Document and source code listings. Any item in the Developmental Configuration may be stored on electronic media.

3.16 Evaluation. See DOD-STD-2168.

3.17 Firmware. The combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device. The software cannot be readily modified under program control.

3.18 Formal Qualification Testing (FQT). A process that allows the contracting agency to determine whether a configuration item complies with the allocated requirements for that item.

3.19 Functional Baseline. See DOD-STD-480.

3.20 Hardware Configuration Item (HWCi). A configuration item for hardware.

3.21 Independent Verification and Validation (IV&V). Verification and validation performed by a contractor or Government agency that is not responsible for developing the product or performing the activity being evaluated. IV&V is an activity that is conducted separately from the software development activities governed by this standard.

3.22 Non-developmental software (NDS). Deliverable software that is not developed under the contract but is provided by the contractor, the Government, or a third party. NDS may be referred to as reusable software, Government furnished software, or commercially available software, depending on its source.

3.23 Product Baseline. See DOD-STD-480.

3.24 Release. A configuration management action whereby a particular version of software is made available for a specific purpose (e.g., released for test).

3.25 Reusable software. Software developed in response to the requirements for one application that can be used, in whole or in part, to satisfy the requirements of another application.

3.26 Software development file (SDF). A repository for a collection of material pertinent to the development or support of software. Contents typically include (either directly or by reference) design considerations and constraints, design documentation and data, schedule and status information, test requirements, test cases, test procedures, and test results.

3.27 Software development library (SDL). A controlled collection of software, documentation, and associated tools and procedures used to facilitate the orderly development and subsequent support of software. The SDL includes the Developmental Configuration as part of its contents. A software development library provides storage of and controlled access to software and documentation in human-readable form, machine-readable form, or both. The library may also contain management data pertinent to the software development project.

3.28 Software engineering environment. The set of automated tools, firmware devices, and hardware necessary to perform the software engineering effort. The automated tools may include but are not limited to compilers, assemblers, linkers, loaders, operating system, debuggers, simulators, emulators, test tools, documentation tools, and data base management system(s).

3.29 Software support. The sum of all activities that take place to ensure that implemented and fielded software continues to fully support the operational mission of the software.

3.30 Software test environment. A set of automated tools, firmware devices, and hardware necessary to test software. The automated tools may include but are not limited to test tools such as simulation software, code analyzers, etc. and may also include those tools used in the software engineering environment.

3.31 System Specification. A system level requirements specification. A system specification may be a System/Segment Specification (SSS), Prime Item Development Specification (PIDS), or Critical Item Development Specification (CIDS).

3.32 Validation. The process of evaluating software to determine compliance with specified requirements.

3.33 Verification. The process of evaluating the products of a given software development activity to determine correctness and consistency with respect to the products and standards provided as input to that activity.

3.34 Version. An identified and documented body of software. Modifications to a version of software (resulting in a new version) require configuration management actions by either the contractor, the contracting agency, or both.

3.35 Definitions of acronyms used in this standard. See Appendix A.

4. GENERAL REQUIREMENTS

4.1 Software development management. The contractor shall perform software development management in compliance with the following requirements.

4.1.1 Software development process. The contractor shall implement a process for managing the development of the deliverable software. The contractor's software development process for each CSCI shall be compatible with the contract schedule for formal reviews and audits. The software development process shall include the following major activities, which may overlap and may be applied iteratively or recursively:

- a. System Requirements Analysis/Design
- b. Software Requirements Analysis
- c. Preliminary Design
- d. Detailed Design
- e. Coding and CSU Testing
- f. CSC Integration and Testing
- g. CSCI Testing.
- h. System Integration and Testing.

4.1.2 Formal reviews/audits. During the software development process, the contractor shall conduct or support formal reviews and audits as required by the contract. Guidance on formal reviews and audits is provided in MIL-STD-1521. The relationship of the formal reviews and audits to software and hardware development is shown in Figure 1. Figure 2 illustrates the occurrence of formal reviews and audits for software and shows the relationship of deliverable products to baselines and the Developmental Configuration.

4.1.3 Software development planning. The contractor shall develop plans for conducting the activities required by this standard. These plans shall be documented in a Software Development Plan (SDP). Following contracting agency approval of the SDP, the contractor shall conduct the activities required by this standard in accordance with the SDP. With the exception of scheduling information, updates to the SDP shall be subject to contracting agency approval.

4.1.4 Risk management. The contractor shall document and implement procedures for risk management. The contractor shall identify, analyze, prioritize, and monitor the areas of the software development project that involve potential technical, cost, or schedule risks.

4.1.5 Security. The contractor shall comply with the security requirements specified in the contract.

4.1.6 Subcontractor management. The contractor shall pass down to the subcontractor(s) all contractual requirements necessary to ensure that all software and associated documentation delivered to the contracting agency are developed in accordance with the prime contract requirements. The contractor shall provide to the subcontractor(s) the baselined requirements for the software to be developed by the subcontractor(s).

4.1.7 Interface with software IV&V agent(s). The contractor shall interface with the software Independent Verification and Validation (IV&V) agent(s) as specified in the contract.

4.1.8 Cost/schedule reporting. The contractor shall prepare and maintain cost/schedule reports at the CSCI level. The cost reports shall indicate budgeted versus actual expenditures and shall conform to the Work Breakdown Structure (WBS) applicable to the development effort. These reports shall also indicate to the contracting agency planned, actual, and predicted progress.

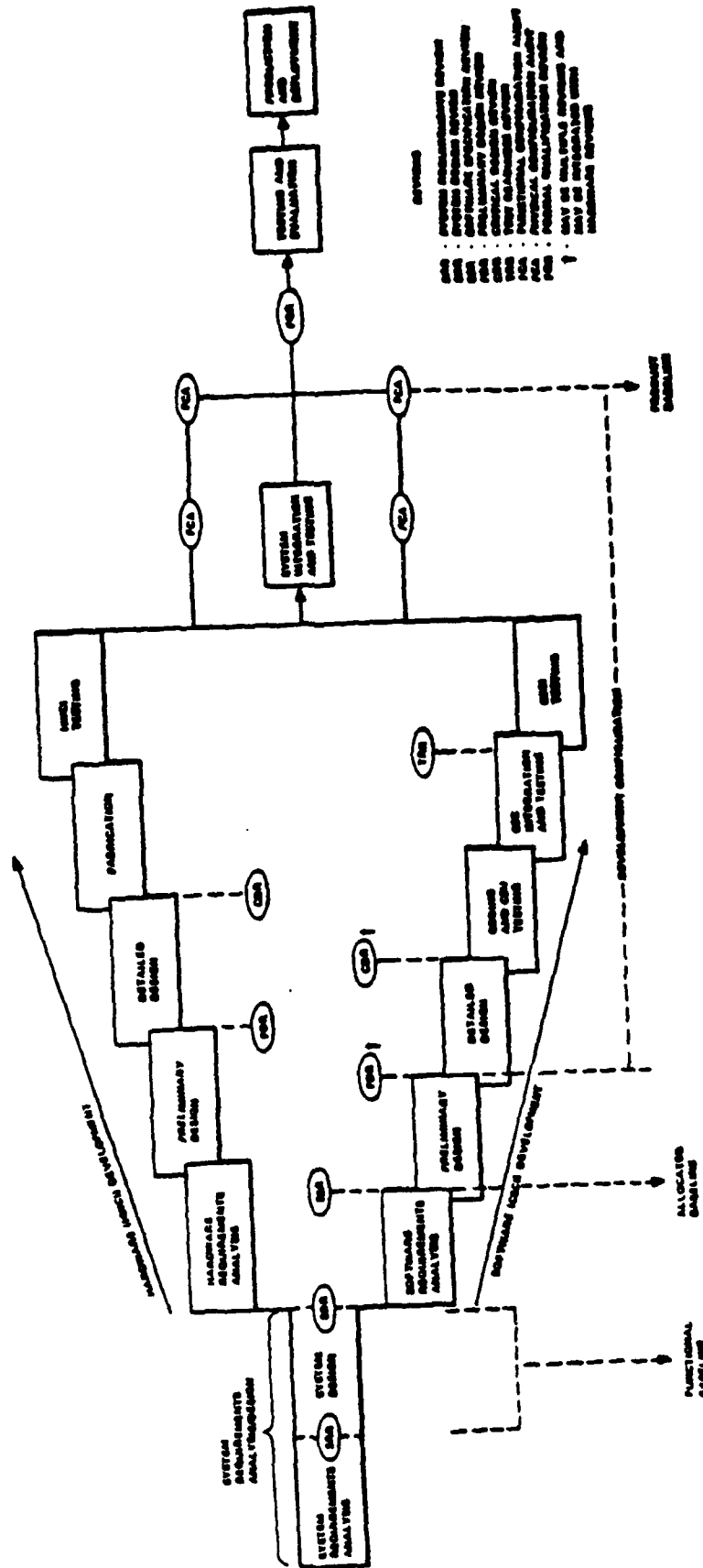


FIGURE 1. An example of system development reviews and audits.

4.1.9 Software development library. The contractor shall establish a software development library (SDL). The contractor shall document and implement procedures for controlling software and associated documentation residing within the SDL. The contractor shall maintain the SDL for the duration of the contract.

4.1.10 Corrective action process. The contractor shall document and implement a corrective action process for handling all problems detected in the products under configuration control and in the software development activities required by the contract. The corrective action process shall comply with the following requirements:

- a. The process shall be closed-loop, ensuring that all detected problems are promptly reported and entered into the corrective action process, action is initiated on them, resolution is achieved, status is tracked and reported, and records of the problems are maintained for the life of the contract.
- b. Inputs to the corrective action process shall consist of problem/change reports and other discrepancy reports.
- c. Each problem shall be classified by category and by priority. The categories and priorities identified in Appendix C shall be included in the category and priority classifications.
- d. Analysis shall be performed to detect trends in the problems reported.
- e. Corrective actions shall be evaluated to: (1) verify that problems have been resolved, adverse trends have been reversed, and changes have been correctly implemented in the appropriate processes and products, and (2) to determine whether additional problems have been introduced.

4.1.11 Problem/change report. The contractor shall prepare a problem/change report to describe each problem detected in software or documentation that has been placed under configuration control. The problem/change report shall describe the corrective action needed and the actions taken to resolve the problem. These reports shall serve as input to the corrective action process.

4.2 Software engineering. The contractor shall perform software engineering in compliance with the following requirements.

4.2.1 Software development methods. The contractor shall use systematic and well documented software development methods to perform requirements analysis, design, coding, integration, and testing of the deliverable software. The contractor shall implement software development methods that support the formal reviews and audits required by the contract.

4.2.2 Software engineering environment. The contractor shall establish a software engineering environment to perform the software engineering effort. The software engineering environment shall comply with the security requirements of the contract. The contractor shall document and implement plans for the installation, configuration control, and maintenance of each item of the environment.

4.2.3 Safety analysis. The contractor shall perform the analysis necessary to ensure that the software requirements, design, and operating procedures minimize the potential for hazardous conditions during the operational mission. Any potentially hazardous conditions or operating procedures shall be clearly identified and documented.

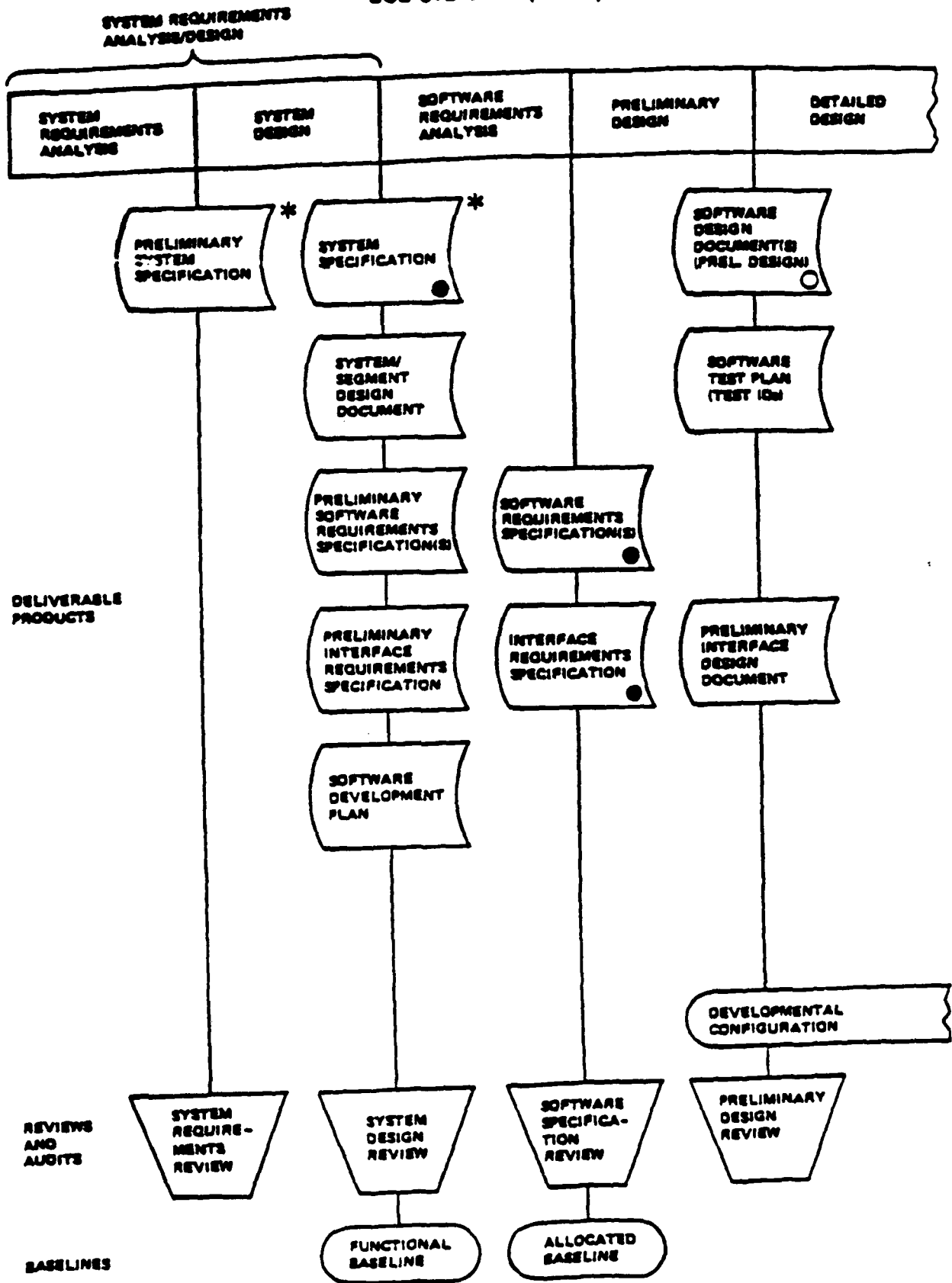


FIGURE 2. Deliverable products, reviews, audits, and baselines.

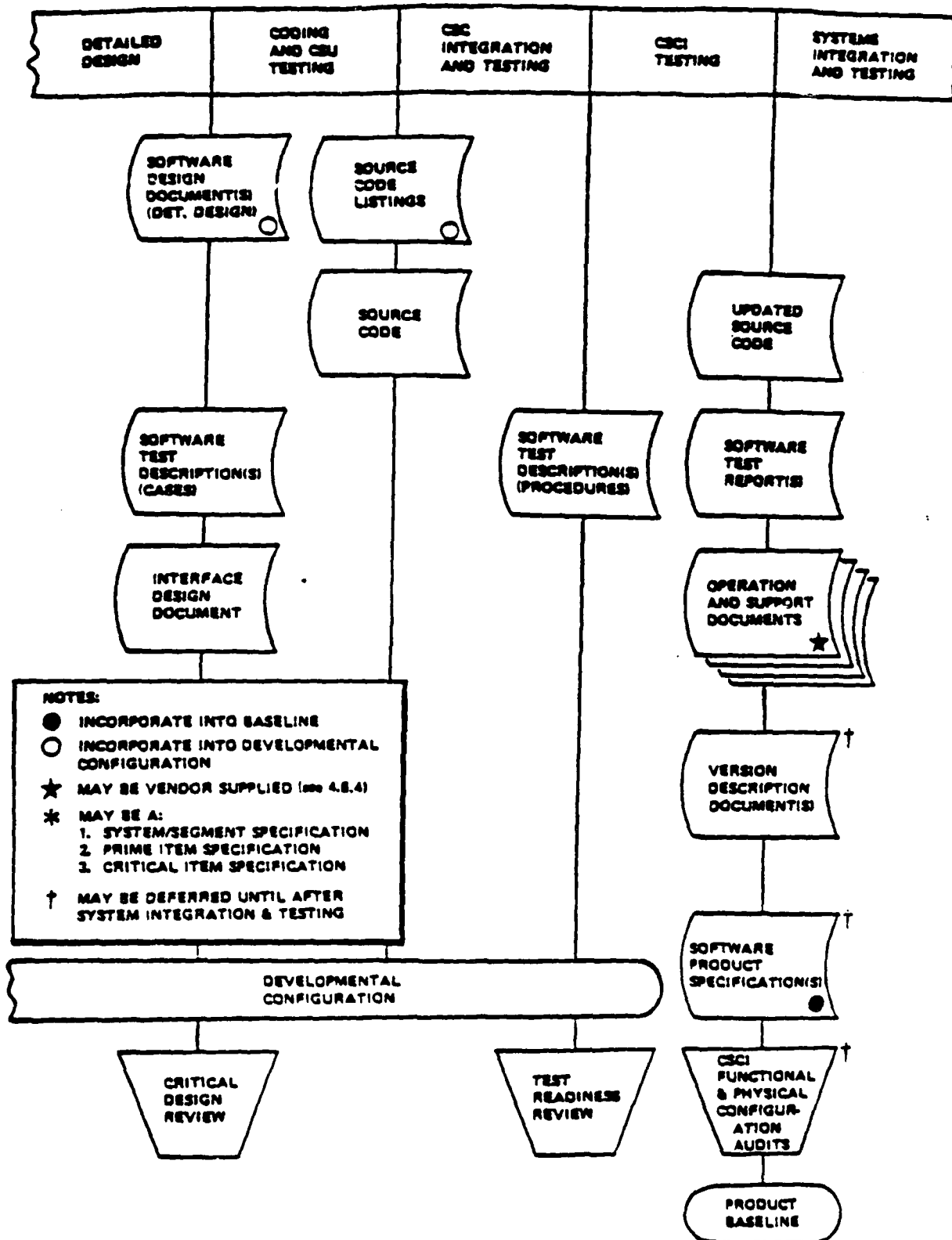


FIGURE 2. Deliverable products, reviews, audits, and baselines - continued.

4.2.4 Non-developmental software. The contractor shall consider incorporating non-developmental software (NDS) into the deliverable software. The contractor shall document plans for using NDS. NDS may be incorporated by the contractor without contracting agency approval only if the NDS is fully documented in accordance with the requirements of this standard. The software development files for NDS need not contain the design considerations, constraints, or data. Incorporation of NDS shall comply with the data rights requirements in the contract.

4.2.5 Computer software organization. The contractor shall decompose and partition each CSCI into Computer Software Components (CSCs) and Computer Software Units (CSUs) in accordance with the development method(s) documented in the Software Development Plan (SDP). The contractor shall ensure that the requirements for the CSCI are completely allocated and further refined to facilitate the design and test of each CSC and CSU. Figure 3 presents an illustration of a system breakdown and CSCI decomposition.

4.2.6 Traceability of requirements to design. The contractor shall document the traceability of the requirements allocated from the system specification to each CSCI, its Computer Software Components (CSCs) and Computer Software Units (CSUs), and from the CSU level to the Software Requirements Specifications (SRSs) and Interface Requirements Specification (IRS).

4.2.7 High order language. The contractor shall use the High Order Language(s) (HOLs) specified in the contract to code the deliverable software. If no HOL is required by the contract, the contractor shall obtain contracting agency approval to use a particular language.

4.2.8 Design and coding standards. The contractor shall document and implement design and coding standards to be used in the development of deliverable software. Software coding standards shall comply with the requirements specified in Appendix B.

4.2.9 Software development files. The contractor shall document the development of each Computer Software Unit (CSU), Computer Software Component (CSC), and CSCI in software development files (SDFs). The contractor shall establish a separate SDF for each CSU or a logically related group of CSUs; each CSC or a logically related group of CSCs; and each CSCI. The contractor shall document and implement procedures for establishing and maintaining SDFs. The contractor shall maintain the SDFs for the duration of the contract. The SDFs shall be made available for contracting agency review upon request. SDFs may be generated, maintained, and controlled by automated means. To reduce duplication, SDFs should not contain information provided in other documents or SDFs. The set of SDFs shall include (directly or by reference) the following information:

- a. Design considerations and constraints
- b. Design documentation and data
- c. Schedule and status information
- d. Test requirements and responsibilities
- e. Test cases, procedures, and results.

4.2.10 Processing resource and reserve capacity. The contractor shall analyze the processing resource and reserve requirements, such as timing, memory utilization, I/O channel utilization, identified in the contract and shall allocate these resources among the CSCIs. The allocation of these resources to a CSCI shall be documented in the Software Requirements Specifications (SRSs) for that CSCI. The contractor shall monitor the utilization of processing resources for the duration of the contract and shall reallocate the resources as necessary to satisfy the reserve requirements. Measured resource utilization at the time of delivery shall be documented in the Software Product Specification (SPS) for each CSCI.

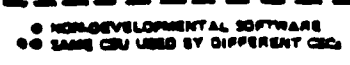


FIGURE 3. Example of a system breakdown and CSCI decomposition.

4.3 Formal qualification testing. The contractor shall conduct FQT of each CSCi on the target computer system or an equivalent system approved by the contracting agency. The contractor's FQT activities shall include stressing the software at the limits of its specified requirements. The contractor may conduct, as part of the FQT activity, testing of CSCis integrated with other CSCis or HWCis that comprise the system.

4.3.1 Formal Qualification Test Planning. The contractor shall develop plans for conducting the formal qualification testing (FQT) activities required by this standard. These plans shall be documented in the Software Test Plan (STP). Following contracting agency approval of the STP, the contractor shall conduct the FQT activities in accordance with the STP. With the exception of scheduling information, updates to the STP shall be subject to contracting agency approval. The contractor shall identify in the Software Test Plan (STP) the tests that involve stressing the software and those that involve integrating CSCis with other configuration items.

4.3.2 Software test environment. The contractor shall establish a software test environment to perform the FQT effort. The software test environment shall comply with the security requirements of the contract. The contractor shall document and implement plans for the installation, test, configuration control, and maintenance of each item of the environment. Following installation, each item of the environment shall be tested to demonstrate that the item performs its intended function.

4.3.3 Independence in FQT activities. The organizations, functions, and persons responsible for fulfilling the FQT requirements of this standard shall have the resources, responsibility, authority, and organizational freedom to permit objective testing and to cause the initiation and verification of corrective action. The persons conducting FQT activities shall not be the persons who developed the software or are responsible for the software. This does not preclude members of the software engineering team from participating in FQT activities.

4.3.4 Traceability of requirements to test cases. The contractor shall document the traceability of the requirements in the Software Requirements Specifications (SRSs) and Interface Requirements Specification (IRS) that are satisfied or partially satisfied by each test case identified in the Software Test Description (STD). The contractor shall document this traceability in the STD for each CSCi.

4.4 Software product evaluations. The contractor shall conduct evaluations of deliverable software and documentation as specified in section 5 of this standard and in compliance with the following requirements.

4.4.1 Independence in product evaluation activities. The organizations, functions, and persons responsible for fulfilling the evaluation requirements of this standard shall have the resources, responsibility, authority, and organizational freedom to permit objective evaluation and to cause the initiation and verification of corrective action. The persons conducting the evaluation of a product shall not be the persons who developed the product or are responsible for the product. This does not preclude members of the development team from participating in these evaluations. Responsibility for the fulfillment of the software product evaluation requirements shall be assigned and specified in the Software Development Plan (SDP).

4.4.2 Final evaluations. Prior to submitting each deliverable item to the contracting agency, the contractor shall internally coordinate the item with appropriate organizations for a final evaluation. The objective of each final evaluation shall be to ensure that the deliverable item is acceptable in terms of its ability to satisfy the need for which it was developed.

4.4.3 Software evaluation records. The contractor shall prepare and maintain records of each software product evaluation performed. When problems have been identified a problem/change report shall be initiated and shall serve as input to the corrective action process. The evaluation records shall be available for contracting agency review and shall be maintained for the life of the contract.

4.4.4 Evaluation criteria. The contractor shall evaluate the products identified in section 5 against the evaluation criteria specified in Figures 4 through 10. Default definitions for the criteria are specified in Appendix D. The contractor may propose additional criteria and alternate definitions for any of the criteria specified in Appendix D. Additional criteria and alternate definitions are subject to contracting agency approval.

4.5 Software configuration management. The contractor shall perform software configuration management in compliance with the following requirements.

4.5.1 Configuration identification. The contractor shall document and implement plans for performing configuration identification. Configuration identification shall be conducted in accordance with the identification scheme specified in the contract. Configuration identification performed by the contractor shall accomplish the following:

- a. Identify the documentation that establishes the Functional, Allocated, and Product Baselines, and the Developmental Configuration.
- b. Identify the documentation and the computer software media containing code, documentation, or both that are placed under configuration control.
- c. Identify each CSCI and its corresponding Computer Software Components (CSCs) and Computer Software Units (CSUs).
- d. Identify the version, release, change status, and any other identification details of each deliverable item.
- e. Identify the version of each CSCI, CSC, and CSU to which the corresponding software documentation applies.
- f. Identify the specific version of software contained on a deliverable medium, including all changes incorporated since its previous release.

4.5.2 Configuration control. The contractor shall document and implement plans for performing configuration control. Configuration control performed by the contractor shall accomplish the following:

- a. Establish a Developmental Configuration for each CSCI.
- b. Maintain current copies of the deliverable documentation and code.
- c. Provide the contracting agency access to documentation and code under configuration control.
- d. Control the preparation and dissemination of changes to the master copies of deliverable software and documentation that have been placed under configuration control so that they reflect only approved changes.

4.5.3 Configuration status accounting. The contractor shall document and implement plans for performing configuration status accounting. The contractor shall generate management records and status reports on all products comprising the Developmental Configuration and the Allocated and Product Baselines. The status reports shall:

- a. Provide traceability of changes to controlled products.
- b. Serve as a basis for communicating the status of configuration identification and associated software.

- c. Serve as a vehicle for ensuring that delivered documents describe and represent the associated software.

4.5.4 Storage, handling, and delivery of project media. The contractor shall document and implement methods and procedures for the storage, handling, and delivery of software and documentation. The contractor shall maintain master copies of the delivered software and documentation.

4.5.5 Engineering Change Proposals. The contractor shall prepare Engineering Change Proposals (ECPs) in accordance with MIL-STD-483, DOD-STD-480, and MIL-STD-481. The contractor shall prepare Specification Change Notices (SCNs) in accordance with MIL-STD-483 and MIL-STD-490.

4.6 Transitioning to software support. The contractor shall provide transition support in compliance with the following requirements.

4.6.1 Regenerable and maintainable code. The contractor shall provide to the contracting agency deliverable code that can be regenerated and maintained using commercially available, Government-owned, or contractually deliverable support software and hardware that has been identified by the contracting agency.

4.6.2 Transition planning. The contractor shall prepare plans for transitioning the deliverable software from development to support. These plans shall be documented in the Computer Resource Integrated Support Document (CRISD).

4.6.3 Software transition and continuing support. The contractor shall perform installation and checkout of the deliverable software in the support environment designated by the contracting agency. The contractor shall provide training and continuing assistance to the contracting agency's support activity as specified in the contract.

4.6.4 Software support and operational documentation. The contractor shall develop and deliver the following software support and operational documentation as required by the Contract Data Requirements List (CDRL):

- a. Computer Resources Integrated Support Document (CRISD)
- b. Computer System Operator's Manual (CSOM)
- c. Software User's Manual (SUM)
- d. Software Programmer's Manual (SPM)
- e. Firmware Support Manual (FSM).

5. DETAILED REQUIREMENTS

5.1 System requirements analysis/design. The contractor shall perform the following system requirements analysis/design activities.

5.1.1 Software development management.

5.1.1.1 The contractor shall support the System Requirements Review (SRR) as specified in the contract.

5.1.1.2 The contractor shall support the System Design Review (SDR) as specified in the contract.

5.1.2 Software engineering.

5.1.2.1 The contractor shall analyze the preliminary system specification and shall determine whether the software requirements are consistent and complete.

5.1.2.2 The contractor shall conduct analysis to determine the best allocation of system requirements between hardware, software, and personnel in order to partition the system into HWCIs, CSCIs, and manual operations. The contractor shall document the allocation in a System/Segment Design Document (SSDD).

5.1.2.3 The contractor shall define a preliminary set of engineering requirements for each CSCI. The contractor shall document these requirements in the preliminary Software Requirements Specification (SRS) for each CSCI.

5.1.2.4 The contractor shall define a preliminary set of interface requirements for each external interface of each CSCI. The contractor shall document these requirements in a preliminary Interface Requirements Specification (IRS).

5.1.3 Formal qualification testing. The contractor shall define a preliminary set of qualification requirements for each CSCI. The contractor shall document these requirements in the preliminary Software Requirements Specification (SRS) for each CSCI. These requirements shall be consistent with the qualification requirements defined in the system specification.

5.1.4 Software product evaluations. The contractor shall perform evaluations of the following products, using the evaluation criteria specified in Figure 4:

- a. The Software Development Plan (SDP)
- b. The System/Segment Design Document (SSDD)
- c. The preliminary Software Requirements Specification (SRS) for each CSCI
- d. The preliminary Interface Requirements Specification (IRS).

5.1.5 Configuration management. The contractor shall place the following documents under configuration control prior to delivery to the contracting agency:

- a. The Software Development Plan (SDP)
- b. The System/Segment Design Document (SSDD)
- c. The preliminary Software Requirements Specification (SRS) for each CSCI.
- d. The preliminary Interface Requirements Specification (IRS).

Evaluation Criteria (See Appendix D)	Item to be Evaluated	Internal Consistency						Notes: Clarification or Additional Criteria
		Understandability	Traceability to the indicated documents	Consistency with the indicated documents	Appropriate analysis, design or coding techniques used	Appropriate allocation of resources and timing resources	Adequate test coverage of requirements	
	System/Segment Design Document (SSDD)	●	●	See Notes				Traceability to system specification and SOW
	Software Development Plan (SDP)	●	●	SOW CDRL				
	Preliminary Software Requirements Specification(s) (SRs)	●	●	See Notes	●	●	●	Traceability to System Specification and SOW Consistency with IRS and other specifications for interfacing items Testability of requirements Adequacy of quality factor requirements
	Preliminary Interfaces Requirements Specification (IRS)	●	●	See Notes	●	●		Traceability to System Specification and SOW Consistency with other specifications for interfacing items Testability of requirements

FIGURE 4. Evaluation criteria for products of system requirements analysis/design.

5.2 Software requirements analysis. The contractor shall perform the following software requirements analysis activities.

5.2.1 Software development management. The contractor shall conduct one or more Software Specification Review(s) (SSR) in accordance with MIL-STD-1521. Following successful completion of an SSR and when authenticated by the contracting agency, the Software Requirements Specifications (SRSs) and associated Interface Requirements Specification (IRS) will establish the Allocated Baseline for the CSCIs.

5.2.2 Software engineering.

5.2.2.1 The contractor shall define a complete set of engineering requirements for each CSCI. The contractor shall document these requirements in the Software Requirements Specification (SRS) for each CSCI.

5.2.2.2 The contractor shall define a complete set of interface requirements for each external interface of each CSCI. The contractor shall document these requirements in an Interface Requirements Specification (IRS).

5.2.3 Formal qualification testing. The contractor shall define a complete set of qualification requirements for each CSCI. The contractor shall document these requirements in the Software Requirements Specification (SRS) for each CSCI.

5.2.4 Software product evaluations. The contractor shall perform evaluations of the products identified below, using the evaluation criteria specified in Figure 5. The contractor shall present a summary of the evaluation results at the Software Specification Review(s).

- a. The Software Requirements Specification (SRS) for each CSCI
- b. The Interface Requirements Specification (IRS).

5.2.5 Configuration management. The contractor shall place the Software Requirements Specification (SRS) for each CSCI and the associated Interface Requirements Specification (IRS) under configuration control prior to delivery to the contracting agency.

Evaluation Criteria (See Appendix D)	Item to be Evaluated	Notes: Clarification or Additional Criteria						
		Internal Consistency	Understandability	Traceability to the indicated documents	Consistency with the indicated documents	Appropriate analysis, design, or coding techniques used	Appropriate allocation of staffing and timing resources	Adequate test coverage of requirements
	Software Requirements Specification(s) (SRSs)	●	●	See Notes	See Notes	●	●	●
	Interface Requirements Specification (IRS)	●	●	See Notes	See Notes	●		
		Traceability to system specification and SOW Consistency with IRS and other specifications for interfacing items Testability of requirements Adequacy of quality factor requirements						
		Traceability to system specification and SOW Consistency with other specifications for interfacing items Testability of requirements						

FIGURE 5. Evaluation criteria for products of Software Requirements Analysis.

5.3 Preliminary design. The contractor shall perform the following preliminary design activities.

5.3.1 Software development management. The contractor shall conduct one or more Preliminary Design Review(s) (PDR) in accordance with MIL-STD-1521.

5.3.2 Software engineering.

5.3.2.1 The contractor shall develop a preliminary design for each CSCl, shall allocate requirements from the Software Requirements Specifications (SRSs) and associated Interface Requirements Specifications (IRS) to the CSCs of each CSCl, and shall establish design requirements for each CSC. The contractor shall document this information in the Software Design Document (SDD) for each CSCl.

5.3.2.2 The contractor shall develop a preliminary design for the external interfaces of each CSCl documented in the Interface Requirements Specification (IRS). The contractor shall document this information in a preliminary Interface Design Document (IDD).

5.3.2.3 The contractor shall document any additional engineering information generated during the preliminary design process in Section 6 of the Software Design Document (SDD) for each CSCl. The engineering information shall include rationale, results of analyses and trade-off studies, and any other information that aids in understanding the preliminary design.

5.3.2.4 The contractor shall establish test requirements for conducting CSC integration and testing. The contractor's CSC integration and testing shall include stressing the software at the limits of its specified requirements. The contractor shall record the test requirements (directly or by reference) in the CSC software development files.

5.3.3 Formal qualification testing. The contractor shall identify the formal qualification tests to be conducted to comply with the qualification requirements identified in the Software Requirements Specification(s) (SRSs). The contractor shall document this information for each CSCl in the Software Test Plan (STP).

5.3.4 Software product evaluations. The contractor shall perform evaluations of the products identified below, using the evaluation criteria specified in Figure 6. The contractor shall present a summary of the evaluation results at the Preliminary Design Review(s).

- a. The Software Design Document (SDD) for each CSCl
- b. The preliminary Interface Design Document (IDD)
- c. The Software Test Plan (STP)
- d. The CSC test requirements.

5.3.5 Configuration management.

5.3.5.1 The contractor shall incorporate the Software Design Document (SDD) for each CSCl into the CSCl's Developmental Configuration prior to delivery to the contracting agency.

5.3.5.2 The contractor shall place the Software Test Plan (STP) under configuration control prior to delivery to the contracting agency.

5.3.5.3 The contractor shall place the preliminary Interface Design Document (IDD) under configuration control prior to delivery to the contracting agency.

Evaluation Criteria (See Appendix D)		Notes: Clarification or Additional Criteria					
Item to be Evaluated	Internal Consistency	Understandability	Traceability to the indicated documents	Consistency with the indicated documents	Appropriate analysis, design or coding techniques used	Appropriate allocation of size and timing resources	Adequate test coverage of requirements
Software Design Document(s) (SDDs) - Preliminary Design	•	•	IRS SRSs	IDD	•	•	Adequacy of requirements allocation for CSC to CSCs
Preliminary Interface Design Document (IDD)	•	•	IRS SRSs	SDDs			
Software Test Plan (STP)	•	•	SSDD IRS SRSs	SDP			Adequacy of data recording, reduction, and analysis methods
CSC test requirements	•	•					

FIGURE 6. Evaluation criteria for products of Preliminary Design.

5.4 Detailed design. The contractor shall perform the following detailed design activities.

5.4.1 Software Development Management. The contractor shall conduct one or more Critical Design Review(s) (CDR) in accordance with MIL-STD-1521.

5.4.2 Software engineering.

5.4.2.1 The contractor shall develop a detailed design for each CSCI, shall allocate requirements from the Computer Software Components (CSCs) to the Computer Software Units (CSUs) of each CSCI, and shall establish design requirements for each CSU. The contractor shall document this information in the Software Design Document (SDD) for each CSCI.

5.4.2.2 The contractor shall develop the detailed design of the CSCI external interfaces documented in the Interface Requirements Specification (IRS). The contractor shall document this information in the Interface Design Document (IDD).

5.4.2.3 The contractor shall document any additional engineering information generated during the detailed design process in Section 6 of the Software Design Document (SDD) for each CSCI. The engineering information shall include rationale, results of analyses and trade-off studies, and any other information that aids in understanding the detailed design.

5.4.2.4 The contractor shall establish test responsibilities, test cases (in terms of inputs, expected results, and evaluation criteria), and schedules for CSC integration and testing. The contractor shall record this information (directly or by reference) in the CSC software development files.

5.4.2.5 The contractor shall establish test requirements, test responsibilities, test cases (in terms of inputs, expected results, and evaluation criteria), and schedules for testing all CSUs. The contractor's CSU testing shall include stressing the software at the limits of its specified requirements. The contractor shall record this information (directly or by reference) in the CSU software development files.

5.4.3 Formal qualification testing. The contractor shall identify and describe the test cases for the formal qualification test identifications in the Software Test Plan (STP). The contractor shall document this information in the Software Test Description (STD) for each CSCI.

5.4.4 Software product evaluations. The contractor shall perform evaluations of the products identified below, using the evaluation criteria specified in Figure 7. The contractor shall present a summary of the evaluation results at the Critical Design Review(s).

- a. The updated Software Design Document (SDD) for each CSCI
- b. The updated Interface Design Document (IDD)
- c. CSC test cases
- d. CSU test requirements and test cases
- e. A specified percentage of the set of CSU and CSC software development files (SDFs). The specified percentage shall be as identified in the Software Development Plan (SDP).
- f. The Software Test Description (STD) for each CSCI.

5.4.5 Configuration management.

5.4.5.1 The contractor shall incorporate the updated Software Design Document (SDD) for each CSCI into the CSCI's Developmental Configuration prior to delivery to the contracting agency.

5.4.5.2 The contractor shall place the updated Interface Design Document (IDD) under configuration control prior to delivery to the contracting agency.

5.4.5.3 The contractor shall place the Software Test Description (STD) for each CSCI under configuration control prior to delivery to the contracting agency.

Evaluation Criteria (See Appendix D)	Item to be Evaluated	Internal Consistency						Notes: Clarification or Additional Criteria
		Understanding	Traceability to the indicated documents	Consistency with the indicated documents	Appropriate analysis, design or coding techniques used	Appropriate allocation of staffing and timing resources	Adequate test coverage of requirements	
Software Design Document(s) (SDDs) - Detailed Design		•	•	IRS SRSs	IDD	•	•	Consistency between data definition and data use Accuracy and required precision of constants
	Interface Design Document (IDD)	•	•	IRS SRSs	SDDs			
CSU test requirements and test cases		•	•		SDDs IDD		•	Adequate detail in specifying test inputs, expected results, and evaluation criteria
CSC test cases		•	•	IRS SRSs	SDDs IDD		•	Adequate detail in specifying test inputs, expected results, and evaluation criteria
Contents of CSU and CSC SDFs		•	•	See Notes	See Notes			Traceability of CSU SDFs to CSC SDFs
Software Test Descriptions (STDs) - Test cases		•	•	IRS SRSs			•	Adequate detail in specifying test inputs expected results, and evaluation criteria

FIGURE 7. Evaluation criteria for products of Detailed Design.

5.5 Coding and CSU testing. The contractor shall perform the following Coding and CSU testing activities.

5.5.1 Software development management. No additional requirements.

5.5.2 Software engineering.

5.5.2.1 The contractor shall develop test procedures for conducting each CSU test. The contractor shall record these procedures in the corresponding CSU software development files (SDFs).

5.5.2.2 The contractor shall code and test each CSU ensuring that the algorithms and logic employed by each CSU are correct and that the CSU satisfies its specified requirements. The contractor shall record the test results of all CSU testing in the corresponding CSU SDFs.

5.5.2.3 The contractor shall make all necessary revisions to the design documentation and code, perform all necessary retesting, and shall update the SDFs of all CSUs that undergo design or coding changes based on CSU tests.

5.5.2.4 The contractor shall develop test procedures for conducting each CSC test. The contractor shall record these procedures in the CSC SDFs.

5.5.3 Formal qualification testing. No additional requirements.

5.5.4 Software product evaluations. The contractor shall perform evaluations of the products identified below, using the evaluation criteria specified in Figure 8.

- a. The source code for each CSU
- b. The CSC test procedures
- c. The CSU test procedures and test results
- d. A specified percentage of the set of updated software development files (SDFs).

5.5.5 Configuration management.

5.5.5.1 The contractor shall incorporate the updated Software Design Documents (SDDs) and source code listings for each successfully tested and evaluated CSU into the appropriate Developmental Configuration.

5.5.5.2 The contractor shall place the source code for each successfully tested and evaluated CSU under configuration control.

FIGURE 8. Evaluation criteria for products of Coding and CSU Teating.

FIGURE 8. Evaluation criteria for products of Coding and CSU Teating.

5.6 CSC integration and testing. The contractor shall perform the following CSC integration and testing activities.

5.6.1 Software development management. The contractor shall conduct one or more Test Readiness Review(s) (TRR) in accordance with MIL-STD-1521.

5.6.2 Software engineering.

5.6.2.1 The contractor shall conduct CSC integration and testing. The contractor shall ensure that the algorithms and logic employed by each CSC are correct and that the CSC satisfies its specified requirements.

5.6.2.2 The contractor shall record the test results of all CSC integration and testing in the corresponding CSC software development files (SDFs).

5.6.2.3 The contractor shall make all necessary revisions to the design documentation and code, perform all necessary retesting, and update the software development files (SDFs) of all CSUs, CSCs, and CSCIs that undergo design or coding changes based on the results of all testing performed.

5.6.3 Formal qualification testing.

5.6.3.1 For each formal qualification test case identified in the Software Test Description(s) (STDs) the contractor shall develop set-up procedures, procedures for conducting each test, and procedures for analyzing the test results. These procedures shall be documented in the Software Test Description (STD) for each CSCI.

5.6.3.2 Prior to conducting Formal Qualification Testing (FQT), the contractor shall conduct the tests documented in the Software Test Description (STD) for each CSCI to ensure that the procedures are complete and accurate and that the software is ready for FQT. The contractor shall record the results of this activity in the corresponding CSCI software development files (SDFs) and shall update the STD as appropriate.

5.6.4 Software product evaluations. The contractor shall perform evaluations of the products identified below, using the evaluation criteria specified in Figure 9. The contractor shall present a summary of the evaluation results at the Test Readiness Review.

- a. The test results recorded in the software development files (SDFs)
- b. The updated Software Test Description (STD) for each CSCI
- c. The updated source code and design documents
- d. A specified percentage of the updated software development files (SDFs).

5.6.5 Configuration management. The contractor shall incorporate the updated Software Design Documents (SDDs) and source code listings for each successfully tested and evaluated CSC into the appropriate Developmental Configuration.

31

FIGURE 9. Evaluation criteria for products of CSC Integration and Testing.

5.7 CSCI testing. The contractor shall perform the following CSCI testing activities.

5.7.1 Software development management. The contractor shall support the Functional Configuration Audit(s) (FCA) and Physical Configuration Audit(s) (PCA). The FCA and PCA for a CSCI may be delayed until after system integration and testing.

5.7.2 Software engineering.

5.7.2.1 The contractor shall make necessary revisions to the Software Design Document(s) (SDDs) and code, conduct all necessary retesting, and update the software development files (SDFs) of all CSUs, CSCs, and CSCIs that undergo design or coding changes based on the results of formal qualification testing.

5.7.2.2 The contractor shall make necessary revisions to the Interface Design Document (IDD) based on the results of formal qualification testing and shall prepare the IDD for delivery.

5.7.2.3 Following successful completion of formal qualification testing and prior to Functional Configuration Audit (FCA) and Physical Configuration Audit (PCA), the contractor shall produce the updated source code for each CSCI. The contractor shall prepare the source code for each CSCI for delivery as specified in the Software Requirements Specification (SRS).

5.7.2.4 For each CSCI, the contractor shall prepare a Software Product Specification (SPS) for delivery.

5.7.3 Formal qualification testing.

5.7.3.1 The contractor shall perform the formal qualification testing (FQT) activities in accordance with the procedures documented in the Software Test Description (STD) for each CSCI.

5.7.3.2 The contractor shall record the results of formal qualification testing in the Software Test Report(s) (STRs) for each CSCI.

5.7.3.3 Upon completion of FQT, the contractor shall prepare an up-to-date Software Test Description (STD) for delivery to the contracting agency.

5.7.4 Software product evaluations. The contractor shall perform evaluations of the products identified below, using the evaluation criteria specified in Figure 10.

- a. The Software Test Report(s) (STRs) for each CSCI
- b. The updated source code and design documentation.

5.7.5 Configuration management.

5.7.5.1 The contractor shall identify the exact version of each CSCI to be delivered. The contractor shall document this information in a Version Description Document (VDD) for each CSCI.

5.7.5.2 Following successful completion of the Functional Configuration Audit (FCA) and Physical Configuration Audit (PCA) and when authenticated by the contracting agency, the Software Product Specification (SPS) for each CSCI will be incorporated into the Product Baseline. At this point, each CSCI's Developmental Configuration shall cease to exist.

Item to be Evaluated	Evaluation Criteria (See Appendix D)						Notes: Clarification or Additional Criteria
	Internal Consistency	Understandability	Traceability to the indicated documents	Consistency with the indicated documents	Appropriate analysis, design, or coding techniques use	Appropriate allocation of staff and timing resources	Adequate test coverage requirements
Software Test Report(s) (STRs)	●	●	STDs	See Notes			●
Updated source code	●	●	SDDs IDD	SDDs	●	●	
							Conformance to expected results Completeness of testing Completeness of retesting Adequacy of the tested CSCI Compliance with design and coding standards Compliance with maintainability requirements

FIGURE 10. Evaluation criteria for products of CSCI Testing.

5.8 System integration and testing. The contractor shall perform the following System Integration and Testing activities.

5.8.1 Software development management. The contractor shall support the Functional and Physical Configuration Audits (see 5.7.1).

5.8.2 Software engineering. The contractor shall make necessary revisions to design documentation and code and shall perform all retesting necessary based on system integration and testing.

5.8.3 Formal qualification testing.

5.8.3.1 The contractor shall support the development and documentation of system integration and test plans, test cases, and test procedures.

5.8.3.2 The contractor shall support system integration and testing activities.

5.8.3.3 The contractor shall support post test analysis and reporting of system integration and test results.

5.8.4 Software product evaluations. The contractor shall perform evaluations of the updated source code and design documentation using the evaluation criteria specified in Figure 10.

5.8.5 Configuration management. The contractor shall prepare necessary changes to baselined documentation in accordance with paragraph 4.5.5.

6. NOTES

6.1 Data requirements list and cross reference. When this standard is used in an acquisition which incorporates a 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 the DOD FAR Supplement 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.

<u>Paragraph No.</u>	<u>Data Requirements Title</u>	<u>Applicable DID No.</u>
5.1.2.2, 5.1.4, 5.1.5	System/Segment Design Document (SSDD)	DI-CMAN-XXXXX
4.1.3, 4.2.5, 4.4.1, 5.1.4, 5.1.5, 5.4.4, 40.1, 40.2.5	Software Development Plan (SDP)	DI-MCCR-80030
4.2.6, 4.2.10, 4.3.4, 5.1.2.3, 5.1.3, 5.1.4, 5.1.5, 5.2.1, 5.2.2.1, 5.2.3, 5.2.4, 5.2.5, 5.3.2.1, 5.3.3, 5.7.2.3, 40.3.1	Software Requirements Specification (SRS)	DI-MCCR-80025
4.2.6, 4.3.4, 5.1.2.4, 5.1.4, 5.1.5, 5.2.1, 5.2.2.2, 5.2.4, 5.2.5, 5.3.2.1, 5.3.2.2, 5.4.2.2	Interface Requirements Specification (IRS)	DI-MCCR-80026
5.3.2.2, 5.3.4, 5.3.5.3, 5.4.2.2, 5.4.4, 5.4.5.2, 5.7.2.2	Interface Design Document (IDD)	DI-MCCR-80027
5.3.2.1, 5.3.2.3, 5.3.4, 5.3.5.1, 5.4.2.1 5.4.2.3, 5.4.4, 5.4.5.1	Software Design Document (SDD)	DI-MCCR-80012
4.2.10, 5.7.2.4, 5.7.5.2	Software Product Specification (SPS)	DI-MCCR-80029
5.7.5.1	Version Description Document (VDD)	DI-MCCR-80013

<u>Paragraph No.</u>	<u>Data Requirements Title</u>	<u>Applicable DID No.</u>
4.3.1, 5.3.3, 5.3.4, 5.3.5.2, 5.4.3	Software Test Plan (STP)	DI-MCCR-80014
4.3.4, 5.4.3, 5.4.4, 5.4.5.3, 5.6.3.1, 5.6.3.2, 5.6.4, 5.7.3.1, 5.7.3.3	Software Test Description (STD)	DI-MCCR-80015
5.7.3.2, 5.7.4	Software Test Report (STR)	DI-MCCR-80017
4.6.4	Computer System Operator's Manual (CSOM)	DI-MCCR-80018
4.6.4	Software User's Manual (SUM)	DI-MCCR-80019
4.6.4	Software Programmer's Manual (SPM)	DI-MCCR-80021
4.6.4	Firmware Support Manual (FSM)	DI-MCCR-80022
4.6.2, 4.6.4	Computer Resources Integrated Support Document (CRISD)	DI-MCCR-80024

(Data item descriptions related to this standard, and identified in section 6 will be approved and listed as such in DoD 5000.19-L, Vol. II, AMSDL. Copies of data item descriptions required by the contractors in connection with specific acquisition functions should be obtained from the Naval Publications and Forms Center or as directed by the contracting officer.)

6.2 Subject term (key word) listing.

Acquisition
 Baselines
 Code
 Coding and CSU testing
 Computer
 Computer resources
 Computer software
 Computer software component
 Computer software configuration item
 Computer software unit
 Configuration item
 Configuration management
 CSC
 CSC integration and testing

CSCI
CSCI testing
CSU
Data item descriptions
Detailed design
Developmental configuration
Engineering environment
Firmware
Formal Qualification Testing
Non-developmental software
Preliminary design
Qualification
Requirements analysis
Risk management
Safety management
Software
Software development
Software development file
Software development library
Software engineering
Software product evaluation
Software requirements analysis
Software support
System integration and testing
Test environment
Testing

APPENDIX A

LIST OF ACRONYMS AND ABBREVIATIONS

10.1 Purpose. This appendix provides a list of all acronyms and abbreviations used in this standard, with the associated meaning.

10.2 Acronyms.

CDR	Critical Design Review
CDRL	Contract Data Requirements List
CIDS	Critical Item Development Specification
CRISD	Computer Resources Integrated Support Document
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSOM	Computer System Operator's Manual
CSU	Computer Software Unit
DID	Data Item Description
DOD	Department of Defense
DODISS	Department of Defense Index of Specifications and Standards
ECP	Engineering Change Proposal
FAR	Federal Acquisition Regulation
FCA	Functional Configuration Audit
FSM	Firmware Support Manual
FQT	Formal Qualification Testing
GFS	Government Furnished Software
HOL	High order language
HWCI	Hardware Configuration Item
IDD	Interface Design Document
I/O	Input/Output
IRS	Interface Requirements Specification
IV&V	Independent Verification and Validation
NDS	Non-developmental software
PCA	Physical Configuration Audit
PDR	Preliminary Design Review
PIDS	Prime Item Development Specification
SCN	Specification Change Notice
SDD	Software Design Document
SDF	Software development file
SDL	Software development library
SDP	Software Development Plan
SDR	System Design Review
SOW	Statement of Work
SPM	Software Programmer's Manual
SPS	Software Product Specification
SRR	System Requirements Review
SRS	Software Requirements Specification
SSDD	System/Segment Design Document
SSR	Software Specification Review
SSS	System/Segment Specification
STD	Software Test Description
STP	Software Test Plan
STR	Software Test Report

SUM
TRR
VDD
WBS

Software User's Manual
Test Readiness Review
Version Description Document
Work Breakdown Structure

APPENDIX B

REQUIREMENTS FOR SOFTWARE CODING STANDARDS

20.1 Purpose. The purpose of this appendix is to specify language independent requirements for software coding standards.

20.2 Applicability. This appendix applies to all deliverable source code developed under the contract.

20.3 Rules and Conventions. The following subparagraphs define the requirements for rules and conventions applicable to software coding standards. The contractor shall implement software coding standards that comply with these requirements.

20.3.1 Presentation style. The coding standards shall describe the rules and conventions for the format of the source code which may include paper listings, listings stored on electronic media, or both. The rules and conventions for presentation style shall include standards for:

- a. Indentation and spacing
- b. The use of capitalization
- c. Uniform presentation of information throughout the source code (e.g., the grouping together of all data declarations)
- d. Use of headers
- e. Layout of source code listings
- f. Conditions under which comments are provided and the format to be used
- g. Size of code aggregates (e.g. on the average 100 or at most 200 executable, non-expandable statements).

20.3.2 Naming. The coding standards shall describe the rules and conventions governing the selection of identifiers used in the source code listings (e.g., identifiers for CSUs, variables, parameters, packages, procedures, subunits, and other aggregates of code.) Restrictions on the use of reserve words and keywords shall be identified.

20.3.3 Restrictions on the implementation language. The coding standards shall include a description of any restrictions imposed on the use of constructs and features of the implementation language due to project or machine-dependent characteristics. Machine-dependent characteristics may include input/output features, word length-dependent features, use of floating point arithmetic, etc. Project characteristics may include, but are not limited to, safety or security considerations in the operational environment.

20.3.4 Use of language constructs and features. The coding standards shall address the allowed use of constructs and features of the implementation language. For example, when Ada is the implementation language, the coding standards shall address the use of exception handling, goto and abort statements, unchecked conversion, etc.

20.3.5 Complexity. The coding standards shall describe controls and restrictions on the complexity of code aggregates.

APPENDIX C

CATEGORY AND PRIORITY CLASSIFICATIONS
FOR SOFTWARE PROBLEM REPORTING

30.1 Purpose. This appendix contains a category and priority classification scheme to be applied to all problems detected in the deliverable software that has been placed under contractor configuration control.

30.2 Classification by category. Problems detected during software operation shall be classified by category as follows:

- a. **Software problem.** The software does not operate according to supporting documentation and the documentation is correct.
- b. **Documentation problem.** The software does not operate according to supporting documentation but the software operation is correct.
- c. **Design problem.** The software operated according to supporting documentation but a design deficiency exists. The design deficiency may not always result in a directly observable operational symptom but possesses the potential for creating further problems.

30.3 Classification by priority. Problems detected in the software shall be classified by priority as follows:

- a. **Priority 1.** A software problem that does one of the following:
 - (1) Prevents the accomplishment of an operational or mission essential capability specified by baselined requirements
 - (2) Prevents the operator's accomplishment of an operational or mission essential capability
 - (3) Jeopardizes personnel safety.
- b. **Priority 2.** A software problem that does one of the following:
 - (1) Adversely affects the accomplishment of an operational or mission essential capability specified by baselined requirements so as to degrade performance and for which no alternative work-around solution is known
 - (2) Adversely affects the operator's accomplishment of an operational or mission essential capability specified by baselined requirements so as to degrade performance and for which no alternative work-around solution is known.
- c. **Priority 3.** A software problem that does one of the following:
 - (1) Adversely affects the accomplishment of an operational or mission essential capability specified by baselined requirements so as to degrade performance and for which an alternative work-around solution is known
 - (2) Adversely affects the operator's accomplishment of an operational or mission essential capability specified by baselined requirements so as to degrade performance and for which an alternative work-around solution is known.

DCD-STD-2167A (DRAFT)
Appendix C

- d. Priority 4. A software problem that is an operator inconvenience or annoyance and which does not effect a required operational or mission essential capability.
- e. Priority 5. All other errors.

APPENDIX D

EVALUATION CRITERIA

40.1 Purpose. This appendix contains a default set of definitions for the evaluation criteria appearing in Figures 4 through 10. These definitions shall be implemented by the contractor if an alternative set has not been proposed in the Software Development Plan and accepted by the contracting agency.

40.2 Criteria definitions. The following definitions are listed in the order that the criteria appear in Figures 4 through 10. For convenience, the definitions use the word "document" for the item being evaluated, even though in some instances the item being evaluated may be other than a document.

40.2.1 Internal consistency. Internal consistency as used in this standard means that: (1) no two statements in a document contradict one another, (2) a given term, acronym, or abbreviation means the same thing throughout the document, and (3) a given item or concept is referred to by the same name or description throughout the document.

40.2.2 Understandability. Understandability, as used in this standard, means that: (1) the document uses rules of grammar, capitalization, punctuation, symbols, and notation consistent with those specified in the U. S. Government Printing Office Style Manual, (2) all terms not contained in the U. S. Government Printing Office Style Manual or Merriam-Webster's New International dictionary (latest revision) are defined, (3) standard abbreviations listed in MIL-STD-12 are used, (4) all acronyms and abbreviations not listed in MIL-STD-12 are defined, (5) all acronyms and abbreviations are preceded by the word or term spelled out in full the first time they are used in the document, unless the first use occurs in a table, figure, or equation, in which case they are explained in the text or in a footnote, and (6) all tables, figures, and illustrations are called out in the text before they appear, in the order in which they appear in the document.

40.2.3 Traceability to indicated documents. Traceability as used in this standard means that the document in question is in agreement with a predecessor document to which it has a hierarchical relationship. Traceability has five elements: (1) the document in question contains or implements all applicable stipulations of the predecessor document, (2) a given term, acronym, or abbreviation means the same thing in the documents, (3) a given item or concept is referred to by the same name or description in the documents, (4) all material in the successor document has its basis in the predecessor document, that is, no untraceable material has been introduced, and (5) the two documents do not contradict one another.

40.2.4 Consistency with indicated documents. Consistency between documents, as used in this standard, means that two or more documents that are not hierarchically related are free from contradictions with one another. Elements of consistency are: (1) no two statements contradict one another, (2) a given term, acronym, or abbreviation means the same thing in the documents, and (3) a given item or concept is referred to by the same name or description in the documents.

40.2.5 Appropriate analysis, design, and coding techniques used. The contract may include provisions regarding the requirements analysis, design, and coding techniques to be used. The contractor's Software Development Plan (SDP) describes the contractor's proposed implementation of these techniques. This criterion consists of compliance with the techniques specified in the contract and SDP.

40.2.6 Appropriate allocation of sizing and timing resources. This criterion, as used in this standard, means that: (1) the amount of memory or time allocated to a given element does not exceed documented constraints applicable to that element, and (2) the sum of the allocated amounts for all subordinate elements is within the overall allocation for an item.

DOD-STD-2167A (DRAFT)
Appendix D

40.2.7 Adequate test coverage of requirements. This criterion, as used in this standard, means that: (1) every specified requirement is addressed by at least one test, (2) test cases have been selected for both "average" situation and "boundary" situations, such as minimum and maximum values, (3) "stress" cases have been selected, such as out-of-bounds values, and (4) test cases that exercise combinations of different functions are included.

40.3 Additional criteria. The following definitions apply to criteria that are not self-explanatory and that appear in the NOTES column of Figures 4 through 10. These criteria are not included in each figure, but appear only as appropriate.

40.3.1 Adequacy of quality factors. This criterion applies to the quality factor requirements in the Software Requirements Specification (SRS). Aspects to be considered are: (1) trade-offs between quality factors have been considered and documented, and (2) each quality factor accompanied by a feasible method to be used to evaluate compliance with it is required by the SRS DID.

40.3.2 Testability of requirements. A requirement is considered to be testable if an objective and feasible test can be designed to determine whether the requirement is met by the software.

40.3.3 Consistency between data definition and data use. This criterion applies primarily to design documents. It means that each data element is defined in a way that is consistent with its usage in the software logic.

40.3.4 Adequacy of test cases, test procedures, (test inputs, expected results, evaluation criteria). Test cases and test procedures should specify exactly what inputs to provide, what steps to follow, what outputs to expect, and what criteria to use in evaluating the outputs. If any of these elements are not specified, the test case or test procedure is inadequate.

40.3.5 Completeness of testing. Testing is complete if all test cases and all test procedures have been performed, all results have been recorded, and all acceptance criteria have been met.

40.3.6 Completeness of retesting. Retesting consists of repeating a subset of the test cases and test procedures after software corrections have been made to correct problems found in previous testing. Retesting is considered complete if: (1) all test cases and test procedures that revealed problems in the previous testing have been repeated, their results have been recorded, and the results have met acceptance criteria, and (2) all test cases and test procedures that revealed no problems during the previous testing, but that test functions that are affected by the corrections, have been repeated, their results have been recorded, and the results have met acceptance criteria.

INDEX

- ACCEPTANCE CRITERIA 40.3.5, 40.3.6
 ACQUISITION 1.1, 2.1.2, 6.2
 ALLOCATE, ALLOCATION 4.2.5, 4.2.8, 4.2.10, 5.1.2.2,
 5.3.2.1, 5.4.2.1, 40.2.8
 ALLOCATED BASELINE 3.1, 4.2.10, 5.1.2.1, 5.1.2.2
 see BASELINE
 APPROVAL, APPROVED 4.1.3, 4.2.4, 4.2.7, 4.3, 4.3.1, 4.4.4,
 4.5.2.4, 6.1
 AUDITS 2.1.1, 4.1.1, 4.1.2, Figure 1, 4.2.1, Figure 2, 5.7.1,
 5.7.2.3, 5.7.5.2, 5.8.1
 AUTHENTICATED, AUTHENTICATION 3.2, 5.2.1, 5.7.5.2
 BASELINE, BASELINED 3.3, 4.1.2, 4.1.8, Figure 2, 5.8.5,
 6.2, 30.3.a(1), 30.3.b(1), 30.3.b(2), 30.3.c(1),
 30.3.c(2)
 ALLOCATED 3.1, 4.5.1.a, 4.5.3, 5.2.1
 FUNCTIONAL 3.18, 4.5.1.a, 4.5.3
 PRODUCT 3.28, 4.5.1.a, 4.5.3, 5.7.5.2
 CODE, CODING 3.18, 3.30, 4.1.1.a, 4.2.1, 4.2.7, 4.2.8,
 4.5.1.b, 4.5.2.b, 4.5.2.c, 4.6.1, 5.5, 5.5.2.2, 5.5.2.3,
 5.5.4.a, 5.5.5.1, 5.5.5.2, Figure 6, 5.6.2.3, 5.6.4.c, 5.6.5,
 5.7.2.1, 5.7.2.3, 5.7.4.b, 5.8.2, 5.8.4, 6.2, Appendix B,
 40.2.5
 CODING AND CSU TESTING 5.5
 CODING STANDARDS 4.2.8, Appendix B
 COMPUTER 3.4, 3.10, 3.17, 4.3, 4.5.1.b
 COMPUTER DATA 3.4, 3.5, 3.7, 3.17
 COMPUTER HARDWARE 3.5, 3.6, 3.7
 COMPUTER INSTRUCTIONS 3.4, 3.7, 3.17
 COMPUTER RESOURCES 3.6
 COMPUTER RESOURCES INTEGRATED SUPPORT
 DOCUMENT (CRISD) 4.6.2, 4.6.4.a, 6.1
 COMPUTER SOFTWARE COMPONENT (CSC) 3.8, 4.2.5,
 4.2.9, 4.5.1.c, 5.4.2.1
 COMPUTER SOFTWARE CONFIGURATION ITEM (CSCI)
 3.8, 3.9, 3.13, 3.15, 3.18, 4.1.1, 4.1.1.g, 4.1.8, 4.2.5,
 4.2.6, Figure 3, 4.2.9, 4.2.10, 4.3, 4.3.1, 4.3.4, 4.5.1.c,
 4.5.1.e, 4.5.2.a, 5.1.2.2, 5.1.2.3, 5.1.2.4, 5.1.3, 5.1.4.c,
 5.1.5.c, 5.2.1, 5.2.2.1, 5.2.2.2, 5.2.3, 5.2.4.a, 5.2.5,
 5.3.2.1, 5.3.2.2, 5.3.2.3, 5.3.3, 5.3.4.a, 5.3.5.1, 5.4.2.1,
 5.4.2.2, 5.4.2.3, 5.4.3, 5.4.4.a, 5.4.4.f, 5.4.5.1, 5.4.5.3,
 5.6.2.3, 5.6.3.1, 5.6.3.2, 5.6.4, 5.7, 5.7.1, 5.7.2.1, 5.7.2.3,
 5.7.2.4, 5.7.3.1, 5.7.3.2, 5.7.4.a, 5.7.5.1, 5.7.5.2,
 Figure 10
 COMPUTER SOFTWARE UNIT (CSU) 3.8, 3.11, 4.2.5,
 4.2.9, 4.5.1.c, 5.4.2.1
 COMPUTER SYSTEM OPERATOR'S MANUAL (CSOM)
 4.6.4.b, 6.1
 CONFIGURATION CONTROL 2.1.1, 3.15, 4.1.10, 4.1.11,
 4.2.2, 4.3.2, 4.5.1.b, 4.5.2, 4.5.2.c, 4.5.2.d, 5.1.5, 5.2.5,
 5.3.5.2, 5.3.5.3, 5.4.5.2, 5.4.5.3, 5.5.5.2, 30.1
 CONFIGURATION IDENTIFICATION 3.12, 4.5.1, 4.5.3.b
 CONFIGURATION ITEM
 see COMPUTER SOFTWARE CONFIGURATION ITEM
 CONFIGURATION MANAGEMENT (CM) 2.1, 3.24, 3.34,
 4.5, 5.1.5, 5.2.5, 5.3.5, 5.4.5, 5.5.5, 5.6.5, 5.7.5, 5.8.5
 CONFIGURATION STATUS ACCOUNTING 4.5.3
 CONTRACT, CONTRACTUAL 1.2, 1.3, 4.1.8, 4.8.1
 AS SPECIFIED IN 4.1.5, 4.1.7, 4.2.7, 4.5.1, 4.6.3,
 5.1.1.1, 5.1.1.2
 DURATION OF 4.1.8, 4.2.9, 4.2.10
 IDENTIFIED IN 4.2.10
 IN ACCORDANCE WITH 6.1
 LIFE OF 4.1.10.a, 4.4.3
 REQUIRED BY 4.1.2, 4.1.10, 4.2.1, 4.2.7
 REQUIREMENTS OF 4.1.8, 4.2.2, 4.2.4, 4.3.2
 CONTRACTING AGENCY 1.3, 2.1.2, 3.14, 3.18, 3.22, 3.34,
 3.38, 4.1.3, 4.1.8, 4.1.8, 4.2.4, 4.2.7, 4.2.9, 4.3, 4.3.1,
 4.4.2, 4.4.3, 4.4.4, 4.5.2.e, 4.6.1, 4.6.3, 5.1.5, 5.2.1,
 5.2.5, 5.3.5.1, 5.3.5.2, 5.3.5.3, 5.4.5.1, 5.4.5.2, 5.4.5.3,
 5.7.3.3, 5.7.5.2, 40.1
 CORRECTIVE ACTION, CORRECTIVE ACTION PROCESS
 4.1.10, 4.1.11, 4.3.3, 4.4.1, 4.4.3
 COST/SCHEDULE REPORTING 4.1.8
 CRITICAL DESIGN REVIEW (CDR) 5.4.1, 5.4.4
 CRITICAL ITEM DEVELOPMENT SPECIFICATION (CIDS)
 3.31
 CSC INTEGRATION AND TESTING 4.1.1.f, 5.3.2.4, 5.4.2.4,
 5.4.4, 5.5.2.4, 5.5.4, 5.6, Figure 6
 CSCI TESTING 4.1.1.g, 4.3, 5.7, Figure 10
 CSU TESTING 4.1.1.e, 5.4.2.5, 5.4.4, 5.5.4, 5.5, Figure 8
 DATA RIGHTS 4.2.4
 DD FORM 1423 6.1
 DD FORM 1684 6.1
 DELIVER, DELIVERABLE, DELIVERED 3.22, 4.1.1, 4.1.2,
 4.1.8, 4.2.1, Figure 2, 4.2.4, 4.2.7, 4.2.8, 4.2.10, 4.4,
 4.4.2, 4.5.1.d, 4.5.1.f, 4.5.2.b, 4.5.2.d, 4.5.3.c, 4.5.4,
 4.6.1, 4.6.2, 4.6.3, 4.6.4, 5.1.5, 5.2.5, 5.3.5.1, 5.3.5.2,
 5.3.5.3, 5.4.5.1, 5.4.5.2, 5.4.5.3, 5.7.2.2, 5.7.2.3, 5.7.2.4,
 5.7.3.3, 5.7.5.1, 6.1, 20.2, 30.1
 DESIGN AND CODING STANDARDS 4.2.8, Appendix B
 DEVELOPMENTAL CONFIGURATION 3.15, 3.27, 4.1.2,
 4.5.1.a, 4.5.2.a, 4.5.3, 5.3.5.1, 5.4.5.1, 5.5.5.1, 5.6.5,
 5.7.5.2
 DETAILED DESIGN 4.1.1.d, 5.4, Figure 7
 DOD-STD-480 2.1.1
 DOD-STD-2168 2.1.1
 ENGINEERING CHANGE PROPOSAL (ECP) 2.1.1, 4.5.5
 ENGINEERING ENVIRONMENT 3.28, 3.30, 4.2.2
 EVALUATE, EVALUATIONS 3.18, 3.21, 3.32, 3.33, 4.1.10.e,
 4.4, 4.4.1, 4.4.2, 4.4.3, 4.4.4, 5.1.4, 5.2.4, 5.3.4, 5.4.2.4,
 EVALUATE, EVALUATIONS 5.4.2.5, 5.4.4, 5.5.4, 5.5.5.1,
 5.5.5.2, 5.6.4, 5.6.5, 5.7.4, 5.8.4, Figures 4-10, Appendix D

INDEX

- EVALUATION CRITERIA 4.4.4, 5.1.4, Figure 4, 5.2.4,
Figure 5, 5.3.4, Figure 6, 5.4.2.4, 5.4.2.5, 5.4.4, Figure
7, 5.5.4, Figure 8, 5.8.4, Figure 9, 5.7.4, Figure 10,
5.8.4, Appendix D
- EVALUATION RECORDS 4.4.3
- EXTERNAL INTERFACE 5.1.2.4, 5.2.2.2, 5.3.2.2, 5.4.2.2
- FEDERAL ACQUISITION REGULATION (FAR) 6.1
- FINAL EVALUATION 4.4.2
- FIRMWARE 1.2.3, 3.17, 3.28, 3.30
- FIRMWARE SUPPORT MANUAL (FSM) 4.8.4.a, 6.1
- FORMAL QUALIFICATION
 - REQUIREMENTS 5.1.3, 5.2.3, 5.3.3
 - ACTIVITIES 4.3, 4.3.1, 4.3.2, 4.3.3, 5.7.3.1
 - TESTING (FQT) 3.18, 4.3, 4.3.1, 5.1.3, 5.2.3, 5.3.3.,
5.4.3, 5.8.3, 5.8.3.1, 5.8.3.2, 5.7.2.1, 5.7.2.2, 5.7.2.3,
5.7.3.2, 5.7.3.3, 5.8.3
- FUNCTIONAL CONFIGURATION AUDIT (FCA) 5.7.1,
5.7.2.3, 5.7.5.2, 5.8.1
- GOVERNMENT FURNISHED SOFTWARE (GFS) 3.22
- HIGH ORDER LANGUAGE (HOL) 4.2.7
- HARDWARE CONFIGURATION ITEM (HWCI) 3.20, 4.3,
5.1.2.2
- INDEPENDENCE 4.3.3, 4.4.1
- INDEPENDENT VERIFICATION AND VALIDATION (IV&V)
3.21, 4.1.7
- INSTALLATION AND CHECKOUT 4.2.2, 4.3.2, 4.8.3
- INTERFACE DESIGN DOCUMENT (IDD) 5.3.2.2, 5.3.4.b,
5.3.5.3, 5.4.2.2, 5.4.4.b, 5.4.5.2, 5.7.2.2, 6.1
- INTERFACE REQUIREMENTS SPECIFICATION (IRS) 4.2.6,
4.3.4, 5.1.2.4, 5.1.4.d, 5.1.5.d, 5.2.1, 5.2.2.2, 5.2.4.b,
5.2.5, 5.3.2.1, 5.3.2.2, 5.4.2.2, 6.1
- I/O CHANNEL UTILIZATION 4.2.10
- INTEGRATION 4.2.1, 4.3, 4.3.1
 - see CSC INTEGRATION AND TESTING
 - see SYSTEM INTEGRATION AND TESTING
- LANGUAGE 20.1, 20.3.4, 20.3.5
- LISTINGS 3.10, 3.15, 5.5.5.1, 5.6.5, 20.3.1
- MANAGEMENT RECORDS 4.5.3
- MASTER COPIES 4.5.2.d, 4.5.4
- MEMORY 4.2.10, 40.2.6
- METHODS/METHODOLOGY 4.5.4, 40.3.1
 - DEVELOPMENT 4.2.1, 4.2.5
- MIL-STD-12 40.2.2
- MIL-STD-481 2.1.1, 4.5.5
- MIL-STD-483 2.1.1, 4.5.5
- MIL-STD-490 2.1.1, 4.5.5
- MIL-STD-499 1.2.1, 2.1.1
- MIL-STD-1521 2.1.1, 4.1.2, 5.2.1, 5.3.1, 5.4.1, 5.6.1
- NON-DEVELOPMENTAL SOFTWARE (NDS) 3.22, 4.2.4
- OPERATING
 - INSTRUCTIONS 3.10
- PROCEDURES 4.2.3
- SYSTEMS 3.28
- OPERATIONAL
 - CAPABILITY 30.3
 - DOCUMENTATION 4.8.4
 - ENVIRONMENT 20.3.3
 - LIFE 3.10
 - MISSION 3.28, 4.2.3
 - REQUIREMENTS 4.2.5
 - SYMPTOM 30.2.c
- ORGANIZATIONAL
 - FREEDOM 4.3.3, 4.4.1
- PHYSICAL CONFIGURATION AUDIT (PCA) 5.7.1, 5.7.2.3,
5.7.5.2, 5.8.1
- PLAN(S), PLANNING 4.1.3, 4.2.2, 4.2.4, 4.3.1, 4.3.2, 4.4.1,
4.5.1, 4.5.2, 4.5.3, 4.8.2, 5.1.4.a, 5.1.5.a, 5.3.3, 5.3.4.c,
5.3.5.2, 5.4.3, 5.4.4.a, 5.8.3.1, 6.1, 40.1, 40.2.5
- PRELIMINARY DESIGN 4.1.1.a, 5.3, Figure 6
- PRELIMINARY DESIGN REVIEW (PDR) 5.3.1, 5.3.4
- PRIME ITEM DEVELOPMENT SPECIFICATION (PIDS) 3.31
- PROBLEM/CHANGE REPORT 4.1.10.b, 4.1.11, 4.4.3
 - CATEGORY 4.1.10.a, 4.1.11, Appendix C, 30.2
 - PRIORITY 4.1.10.a, 4.1.11, Appendix C, 30.3
- PROCEDURES 3.27, 4.1.4, 4.1.9, 4.2.3, 4.2.9, 4.5.4, 5.8.3.1,
20.3.2
 - TEST 3.28, 4.2.9.a, 5.5.2.1, 5.5.2.4, 5.5.4.b, 5.5.4.c,
5.6.3.1, 5.8.3.2, 5.7.3.1, 5.8.3.1, 40.3.4, 40.3.5,
40.3.6
- PRODUCT BASELINE see BASELINE
- PRODUCT EVALUATION 3.18, 3.33, 4.1.10.b, 4.4, 4.4.1,
4.4.3, 4.4.4, 5.1.4, 5.2.4, 5.3.4, 5.4.4, 5.5.4, 5.6.4, 5.7.4,
5.8.4, Figures 4 through 10
- QUALITY FACTORS 40.3.1
- QUALIFICATION 4.3.1, 5.1.3, 5.2.3, 5.3.3
 - see FORMAL QUALIFICATION TESTING
- RELEASE 3.24, 4.5.1.d, 4.5.1.f
- REQUIREMENTS 1.1, 1.3, 3.10, 3.25, 3.32, 4.1, 4.1.6,
4.1.10, 4.2, 4.2.1, 4.2.3, 4.2.4, 4.2.8, 4.3, 4.5, 4.6,
5.1.2.1, 5.4.2.5, 5.5.2.5, 5.6.2.1, 40.2.5, 40.2.7, 40.3.2
- ALLOCATED 3.18, 4.2.5, 4.2.6, 5.1.2.2, 5.3.2.1, 5.4.2.1
- BASELINED 4.1.6, 30.3.a, 30.3.b, 30.3.c
- CODING STANDARDS Appendix B
- DATA RIGHTS 1.2, 4.6.4, 6.1
- DESIGN 5.3.2.1, 5.4.2.1
- ENGINEERING 5.2.2.1
- EVALUATION 4.4.1
- INTERFACE 5.1.2.4, 5.2.2.2
- QUALIFICATION 5.1.3, 5.2.3, 5.3.3
- QUALITY FACTOR 40.3.1
- RESERVE 4.2.10
- SECURITY 4.1.5, 4.2.2, 4.3.2

INDEX

REQUIREMENTS (continued)

SYSTEM 2.21, 4.1.1.a, 5.1, 5.1.2.2
 TEST 3.28, 4.2.2.4, 4.3.3, 5.1.3, 5.2.3, 5.3.2.4, 5.3.3,
 5.3.4, 5.4.2.5, 5.4.4
 see INTERFACE REQUIREMENTS SPECIFICATION
 see SOFTWARE REQUIREMENTS SPECIFICATION
 see TRACEABILITY
 REQUIREMENTS ANALYSIS 4.2.1, 40.2.5
 see SOFTWARE REQUIREMENTS ANALYSIS
 see SYSTEM REQUIREMENTS ANALYSIS/DESIGN
 RESERVE CAPACITY 4.2.10
 RESOURCES 4.3.3, 4.4.1
 COMPUTER 3.8
 PROCESSING 4.2.10
 SIZING AND TIMING 40.2.6
 RETESTING 5.5.2.3, 5.8.2.3, 5.7.2.1, 5.8.2, 40.3.6
 REUSABLE 3.22, 3.25
 REVIEWS 2.1.1, 4.1.1, 4.1.2, Figure 1, 4.2.1, Figure 2,
 5.1.1.1, 5.1.1.2, 5.2.1, 5.2.4, 5.3.1, 5.3.4, 5.4.1,
 5.4.4, 5.8.1, 5.8.4
 CONTRACTING AGENCY REVIEW 4.2.9, 4.4.3
 RISK MANAGEMENT 4.1.4
 SAFETY ANALYSIS 4.2.3, 20.3.3, 30.3
 SCHEDULE 3.28, 4.1.4, 4.2.9, 5.4.2.4, 5.4.2.5
 CONTRACT 4.1.1
 REPORTS 4.1.8
 SECURITY 4.1.5, 4.2.2, 4.3.2, 20.3.3
 SOFTWARE DESIGN DOCUMENT (SDD) 5.3.2.1, 5.3.2.3,
 5.3.4.a, 5.3.5.1, 5.4.2.1, 5.4.2.3, 5.4.4.a, 5.4.5.1, 5.7.2.1,
 6.1
 SOFTWARE DEVELOPMENT FILE(S) (SDFs) 3.28, 4.2.4,
 4.2.9, 5.3.2.4, 5.4.2.4, 5.4.2.5, 5.4.4, 5.5.2.1, 5.5.4.c,
 5.6.2.2, 5.6.2.3, 5.6.3.2, 5.6.4.a, 5.6.4.d, 5.7.2.1
 SOFTWARE DEVELOPMENT LIBRARY (SDL) 3.27, 4.1.9
 SOFTWARE DEVELOPMENT MANAGEMENT 4.1, 5.1.1,
 5.2.1, 5.3.1, 5.4.1, 5.5.1, 5.6.1, 5.7.1, 5.8.1
 SOFTWARE DEVELOPMENT PLAN (SDP) 4.1.3, 4.2.5,
 4.4.1, 5.1.4.a, 5.1.5.a, 5.4.4.e, 6.1, 40.1, 40.2.5
 SOFTWARE DEVELOPMENT PROCESS 4.1.1, 4.1.2
 SOFTWARE ENGINEERING 4.2, 4.3.3, 5.1.2, 5.2.2, 5.3.2,
 5.4.2, 5.5.2, 5.6.2, 5.7.2, 5.8.2
 SOFTWARE ENGINEERING ENVIRONMENT 3.28, 3.30,
 4.2.2
 SOFTWARE PRODUCT EVALUATIONS 4.4, 4.4.1, 4.4.3,
 5.1.4, 5.2.4, 5.3.4, 5.4.4, 5.5.4, 5.6.4, 5.7.4, 5.8.4,
 Figures 4-10
 SOFTWARE PRODUCT SPECIFICATION (SPS) 4.2.10,
 5.7.2.4, 5.7.5.2, 6.1
 SOFTWARE PROGRAMMER'S MANUAL (SPM) 4.6.4.d, 6.1
 SOFTWARE QUALITY PROGRAM 1.2.2, 2.1.1

SOFTWARE REQUIREMENTS ANALYSIS 4.1.1.b, 4.2.1,
 4.2.3, 5.2, Figure 6
 SOFTWARE REQUIREMENTS SPECIFICATION (SRS) 4.2.8,
 4.2.10, 4.3.4, 5.1.2.3, 5.1.3, 5.1.4.a, 5.1.5.a, 5.2.1,
 5.2.2.1, 5.2.3, 5.2.4.a, 5.3.2.1, 5.3.3, 5.7.2.3, 6.1, 40.3.1
 SOFTWARE SPECIFICATION REVIEW (SSR) 5.2.1, 5.2.4
 SOFTWARE SUPPORT 1.1, 1.2.3, 1.2.4, 3.10, 3.28, 3.27,
 3.29, 4.8, 4.8.1, 4.8.2, 4.8.3, 4.8.4, 30.2
 see COMPUTER SOFTWARE INTEGRATED SUPPORT
 DOCUMENT (CRISD)
 see SUPPORT
 SOFTWARE TEST DESCRIPTION (STD) 4.3.4, 5.4.3, 5.4.4.i,
 5.4.5.3, 5.6.3.1, 5.6.3.2, 5.6.4.b, 5.7.3.1, 5.7.3.3, 6.1
 SOFTWARE TEST ENVIRONMENT 3.30, 4.3.2
 SOFTWARE TEST PLAN (STP) 4.3.1, 5.3.3, 5.3.4.c, 5.3.5.2,
 5.4.3, 6.1
 SOFTWARE TEST REPORT (STR) 5.7.3.2, 5.7.4.a, 6.1
 SOFTWARE USER'S MANUAL (SUM) 4.6.4.c, 6.1
 SOURCE CODE 3.16, 5.5.4.a, 5.5.5.2, 5.6.4.a, 5.7.2.3,
 5.7.4.b, 5.8.4, 20.2, 20.3.1
 SOURCE CODE LISTINGS 5.5.5.1, 5.8.5
 SPECIFICATION CHANGE NOTICE (SCN) 4.5.5
 STATUS REPORTS 4.5.3
 STRESS 4.3, 4.3.1, 5.3.2.4, 5.4.2.5, 40.2.7
 SUBCONTRACTOR 1.2.4, 4.1.8
 SUPPORT
 DOCUMENTATION 4.8.4
 ENVIRONMENT 4.8.3
 PLANNING 4.6.2
 REVIEWS AND AUDITS 4.1.2, 4.2.1, 5.1.1.1, 5.1.1.2,
 5.7.1
 TRANSITION 4.6, 4.6.2
 see SOFTWARE SUPPORT
 SYSTEM ANALYSIS see SYSTEM REQUIREMENTS
 ANALYSIS/DESIGN
 SYSTEM DESIGN REVIEW (SDR) 5.1.1.2
 SYSTEM INTEGRATION AND TESTING 4.1.1.h, 5.7.1, 5.8
 SYSTEM REQUIREMENTS ANALYSIS/DESIGN 4.1.1.a, 5.1
 SYSTEM REQUIREMENTS REVIEW (SRR) 5.1.1.1
 SYSTEM/SEGMENT DESIGN DOCUMENT (SSDD) 5.1.2.2,
 5.1.4.b, 5.1.5, 6.1
 SYSTEM/SEGMENT SPECIFICATION (SSS) 3.31
 SYSTEM SPECIFICATION 3.31, 4.2.6, 5.1.2.1, 5.1.3
 TAILORING 1.3
 TARGET COMPUTER 4.3
 TECHNICAL
 DATA 3.10
 DOCUMENTATION 3.15
 RISK 4.1.4

INDEX

TEST, TESTING 3.11, 4.2.1, 4.3.3, 5.8.3.3, 40.3.2
 CASES 3.28, 4.2.8.a, 4.3.4, 5.4.2.4, 5.4.2.5, 5.4.3,
 5.4.4.c, 5.4.4.d, 5.8.3.1, 5.8.3.1, 40.2.7, 40.3.4
 CONDUCT 5.5.2.2, 5.5.2.4, 5.8.3.2, 40.3.5
 ENVIRONMENT 3.30
 see SOFTWARE TEST ENVIRONMENT
 PROCEDURES 3.28, 4.2.8.a, 5.5.2.1, 5.5.2.4, 5.5.4.b,
 5.5.4.c, 5.8.3.1, 5.8.3.1, 40.3.4, 40.3.5, 40.3.6
 REQUIREMENTS 3.28, 4.2.8.d, 5.3.2.4, 5.3.4.d, 5.4.2.5,
 5.4.4.d
 RESPONSIBILITIES 4.2.8.d, 5.4.2.4, 5.4.2.5
 RESULTS 3.28, 4.2.8.e, 5.4.2.5, 5.5.2.5,
 5.5.4.c, 5.8.2.2, 5.8.2.3, 5.8.3.1, 5.8.4, 5.7.3.2,
 5.8.3.3, 40.3.7
 SCHEDULES 5.4.2.4, 5.4.2.5
 TOOLS 3.28, 3.30
 see CODING AND CSU TESTING
 see CSC INTEGRATION AND TESTING
 see CSCI TESTING
 see FORMAL QUALIFICATION TESTING (FOT)
 see RETESTING
 see SOFTWARE TEST DESCRIPTION
 see SOFTWARE TEST ENVIRONMENT
 see SOFTWARE TEST PLAN
 see SOFTWARE TEST REPORT
 see STRESS
 see SYSTEM INTEGRATION AND TESTING
 TEST READINESS REVIEW (TRR) 5.8.1
 TRACEABILITY 4.5.3, 40.2.3, Figures 4 - 10
 REQUIREMENTS TO DESIGN 4.2.6
 REQUIREMENTS TO TEST CASES 4.3.4
 TRAINING 4.8.3
 VALIDATION 3.21, 3.32, 4.1.7
 VERIFICATION 3.21, 3.32, 4.1.7, 4.3.3, 4.4.1
 VERSION 3.24, 3.34, 4.5.1.d, 4.5.1.e, 4.5.1.f, 5.7.5.1
 VERSION DESCRIPTION DOCUMENT (VDD) 5.7.5.1, 6.1
 WORK BREAKDOWN STRUCTURE (WBS) 4.1.8

Custodians:

Navy - EC
Army - AR
Air Force - 10,28

Review Activities:

Army - AR,SC,CR,MI,AV
Navy - EC,MC,TD,OM,SH,AS
Air Force - 02,10,17,28

Preparing Activity:

Navy - EC

(Project MCCR-0022)

Distribution List for IDA Paper P-2018

NAME AND ADDRESS	NUMBER OF COPIES
Sponsor	
CAPT David Hart SDIO/S/BM The Pentagon Washington, D.C. 20301-7100	5 copies
Other	
Defense Technical Information Center Cameron Station Alexandria, VA 22314	2 copies
LTC Charles Anderson RADDC/CO Griffis AFB, NY 13440	5 copies
Dr. James M. Boyle Math and Computer Science Div. Argonne National Laboratory 9700 S. Cass Ave. Argonne, IL 60439-4844	5 copies
Mr. Mark Crosby SAIC MS T-9-3 1710 Goodridge Dr. McLean, Va 22102	1 copy
COL Jim Graham SDIO/S/SE The Pentagon Washington, D.C. 20301-7100	5 copies
Dr. Jung P. Hong Los Alamos National Laboratory MS K-488 P.O. Box 1663 Los Alamos, NM 87544	5 copies
Dr Virginia P. Kobler US Army Strategic Defense Command DASD-H-SBY P.O. Box 1500 Huntsville, AL 35087-3801	5 copies
Lt. N. Ann Kuo ESD/MD Hanscom AFB, MA 01731	5 copies

Major Frank Maressa
SDIO/S/SA
The Pentagon
Washington, D.C. 20301-7100

5 copies

Mr. Dick Martin
Carnegie-Mellon University
Software Engineering Institute
Pittsburgh, PA 15213

5 copies

Ms. Nancy Mathis
MITRE
4717 University DR., Suite 98
Huntsville, AL 35816

5 copies

Mr. Steve McBurnett
Code 5570
US Naval Research Laboratory
Washington, D.C. 20375-5000

5 copies

LTC Ted Mervosh
ESD/MD
Hanscom AFB, MA 01731

5 copies

LTC John Morrison
ESD/MD
Hanscom AFB, MA 01731

5 copies

Mr. Karl Nyberg
Grebyn Corporation
P.O. Box 1144
Vienna, VA 22180

1 copy

COL Dick Paul
ESD/MD
Hanscom AFB, MA 01731

5 copies

MAJ James Price
SDIO/S
The Pentagon
Washington, D.C. 20301-7100

5 copies

Mr. Ron Raposo
RADDC/COT
Griffis AFB, NY 13440

5 copies

LTC Jon Rindt
SDIO/S/BM
The Pentagon
Washington, D.C. 20301-7100

5 copies

Dr. Albert Small
SRS Technologies
Suite 800
1500 Wilson Boulevard
Arlington, VA 22209

1 copy

MAJ Peter Sowa
SDIO/S/BM
The Pentagon
Washington D.C. 20301-7100

5 copies

Mr. J. R. Southern
USA-SDC
DASD-H-SBD
Huntsville, AL 35807-3801

5 copies

Dr. Stephen Squires
DARPA
1400 Wilson Boulevard
Arlington. VA 22209

1 copy

Mr William Stout
ANSER
1215 Jefferson Davis Hwy., Suite 800
Arlington, VA 22202

5 copies

Mr. H. G. Stuebing
Naval Air Development Center
Code 50C
Warminster, PA 18974

5 copies

Dr. Doyle Thomas
USA SDC
DASD-H-SB
P.O. Box 1500
Huntsville AL 35807

5 copies

Mr. Lawrence A. Tubbs
USA-SDC
DASD-H-SB
P.O. Box 1500
Huntsville, AL 35807

5 copies

COL William Wisdom
SDIO/S/TE
The Pentagon
Washington, D.C. 80914-5001

5 copies

LT Marcia Wye
HQ US SPACE COM/J5B
Peterson AFB, CO 80914-5001

1 copy

CSED Review Panel

Dr. Dan Alpert, Director
Center for Advanced Study
University of Illinois
912 W. Illinois Street
Urbana, Illinois 61801

1 copy

Dr. Barry W. Boehm
TRW Defense Systems Group
MS 2-2304
One Space Park
Redondo Beach, CA 90278

1 copy

Dr. Ruth Davis
The Pymatuning Group, Inc.
2000 N. 15th Street, Suite 707
Arlington, VA 22201

1 copy

Dr. Larry E. Druffel
Software Engineering Institute
Shadyside Place
480 South Aiken Av.
Pittsburgh, PA 15231

1 copy

Dr. C. E. Hutchinson, Dean
Thayer School of Engineering
Dartmouth College
Hanover, NH 03755

1 copy

Mr. A. J. Jordano
Manager, Systems & Software
Engineering Headquarters
Federal Systems Division
6600 Rockledge Dr.
Bethesda, MD 20817

1 copy

Mr. Robert K. Lehto
Mainstay
302 Mill St.
Occoquan, VA 22125

1 copy

Mr. Oliver Selfridge
45 Percy Road
Lexington, MA 02173

1 copy

IDA

General W. Y. Smith, HQ	1 copy
Mr. Seymour Deitchman, HQ	1 copy
Mr. Philip Major, HQ	1 copy
Ms. Karen H. Weber, HQ	1 copy
Dr. Jack Kramer, CSED	1 copy
Dr. Robert I. Winner, CSED	1 copy
Dr. John Salasin, CSED	25 copies
Mr. Howard Cohen, CSED	1 copy
Ms. Audrey A. Hook, CSED	1 copy
Dr. Joseph Linn, CSED	1 copy
Dr. Cathy Jo Linn, CSED	5 copies
Mr. Robert Knapper, CSED	5 copies
Ms. Julia Sensiba, CSED	2 copies
Mr. Richard Waychoff, CSED	5 copies
IDA Control & Distribution Vault	3 copies