

DOD-STD-2167

4 JUNE 1985

SUPERSEDING

DOD-STD-1679A (NAVY)

22 OCTOBER 1983

MIL-STD-1644B (TD)

2 MARCH 1984

MILITARY STANDARD

DEFENSE SYSTEM SOFTWARE DEVELOPMENT



AMSC NO. N3608

AREA MCCR

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

DOD-STD-2167

DEPARTMENT OF DEFENSE
Washington, DC 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 8111, 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 contains requirements for the development of Mission-Critical Computer System software. It establishes a uniform software development process which is applicable throughout the system life cycle. The software development process defines development activities which result in: (1) the generation of different types and levels of software and documentation, (2) the application of development tools, approaches, and methods, and (3) project planning and control. It incorporates practices which have been demonstrated to be cost-effective from a life cycle perspective, based on information gathered by the Department of Defense (DOD) and industry.

2. This standard is intended to be dynamic and responsive to the rapidly evolving software technology field. As such, this standard should be selectively applied and tailored to fit the unique characteristics of each software acquisition program. To ensure that the requirements in this standard are appropriate and responsive to software acquisition needs, users of this standard are encouraged to provide feedback to the Preparing Activity. User experience in terms of benefits, pitfalls, and any other useful information encountered in applying this standard will be most helpful.

3. Data Item Descriptions (DIDs) applicable to this standard are listed in Section 6. When used in conjunction with this standard, these DIDs provide a set of concise and complete documents for recording and communicating information generated as a result of adherence to the requirements specified herein.

CONTENTS

<u>Paragraph</u>	<u>Page</u>
1. SCOPE.....	1
1.1 Purpose.....	1
1.2 Application.....	1
1.2.1 Application to various types of software.....	1
1.2.2 Non-applicability of this standard.....	1
1.2.3 Software developed by Government agencies.....	4
1.3 Tailoring of this standard.....	4
2. REFERENCED DOCUMENTS.....	5
2.1 Government documents.....	5
2.1.1 Specifications, standards, and handbooks.....	5
2.1.2 Other Government documents, drawings, and publications.....	5
2.2 Other publications.....	5
2.3 Order of precedence.....	5
3. DEFINITIONS.....	7
3.1 Allocated Baseline.....	7
3.2 Authentication.....	7
3.3 Baseline.....	7
3.4 Certification.....	7
3.5 Computer data definition.....	7
3.6 Computer software (or software).....	7
3.7 Computer Software Component (CSC).....	7
3.8 Computer Software Configuration Item (CSCI).....	7
3.9 Computer Software Documentation.....	7
3.10 Computer software quality (or software quality).....	7
3.11 Configuration Identification.....	8
3.12 Configuration Item.....	8
3.13 Developmental Configuration.....	8
3.14 Firmware.....	8
3.15 Formal test.....	8
3.16 Functional Baseline.....	8
3.17 Hardware Configuration Item (HWCI).....	8
3.18 Informal test.....	8
3.19 Modular.....	8
3.20 Product Baseline.....	8
3.21 Software development library (SDL).....	8
3.22 Top-down.....	9
3.23 Unit.....	9
4. GENERAL REQUIREMENTS.....	11
4.1 Software development cycle.....	11
4.2 Computer software organization.....	11
4.3 Software quality.....	15
4.4 Use of commercially available, reusable, and Government furnished software.....	15

CONTENTS - Continued

<u>Paragraph</u>	<u>Page</u>	
4.5	Subcontractor control.....	16
4.6	Non-deliverable software, firmware, and hardware.....	16
4.7	Firmware.....	16
4.8	Development methodolgies.....	16
4.9	Security.....	16
4.10	Deliverable Data.....	16
4.11	Deviations and waivers.....	17
5.	DETAILED REQUIREMENTS.....	19
5.1	Software Requirements Analysis.....	19
5.1.1	Activities - Software Requirements Analysis.....	19
5.1.2	Products - Software Requirements Analysis.....	22
5.1.3	Formal Reviews - Software Requirements Analysis..	22
5.1.4	Baselines - Software Requirements Analysis.....	22
5.2	Preliminary Design.....	22
5.2.1	Activities - Preliminary Design.....	22
5.2.2	Products - Preliminary Design.....	26
5.2.3	Formal Reviews - Preliminary Design.....	26
5.2.4	Developmental Configuration - Preliminary Design.....	27
5.3	Detailed Design.....	27
5.3.1	Activities - Detailed Design.....	27
5.3.2	Products - Detailed Design.....	30
5.3.3	Formal Reviews - Detailed Design.....	31
5.3.4	Developmental Configuration - Detailed Design....	31
5.4	Coding and Unit Testing.....	31
5.4.1	Activities - Coding and Unit Testing.....	31
5.4.2	Products - Coding and Unit Testing.....	33
5.4.3	Developmental Configuration - Coding and Unit Testing.....	33
5.5	CSC Integration and Testing.....	34
5.5.1	Activities - CSC Integration and Testing.....	34
5.5.2	Products - CSC Integration and Testing.....	35
5.5.3	Formal Reviews - CSC Integration and Testing.....	36
5.5.4	Developmental Configuration - CSC Integration and Testing.....	36
5.6	CSCI Testing.....	36
5.6.1	Activities - CSCI Testing.....	36
5.6.2	Products - CSCI Testing.....	37
5.6.3	Audits - CSCI Testing.....	38
5.6.4	Baselines - CSCI Testing.....	38
5.6.5	Software acceptance.....	38
5.6.6	Installation and checkout.....	39
5.7	Configuration Management (CM).....	39
5.7.1	Activities - Configuration Management.....	39
5.7.1.1	Configuration identification.....	39
5.7.1.2	Configuration control.....	40
5.7.1.3	Configuration status accounting.....	41

CONTENTS - Continued

<u>Paragraph</u>	<u>Page</u>	
5.7.2	Products - Configuration Management.....	41
5.7.3	Audits - Configuration Management.....	42
5.8	Software Quality Evaluation.....	42
5.8.1	Activities - Software Quality Evaluation.....	42
5.8.1.1	Planning.....	42
5.8.1.2	Internal reviews.....	42
5.8.1.2.1	Evaluation Criteria.....	42
5.8.1.2.2	Internal reviews - all phases.....	43
5.8.1.2.3	Internal review - Software Requirements Analysis.....	43
5.8.1.2.4	Internal review - Preliminary Design.....	43
5.8.1.2.5	Internal review - Detailed Design.....	44
5.8.1.2.6	Internal review - Coding and Unit Testing	45
5.8.1.2.7	Internal review - CSC Integration and Testing.....	46
5.8.1.2.8	Internal review - CSCI Testing.....	46
5.8.1.3	Formal reviews and audits.....	47
5.8.1.4	Acceptance inspection.....	47
5.8.1.5	Installation and checkout.....	48
5.8.1.6	Evaluation of subcontractor products.....	48
5.8.1.7	Commercially available, reusable, and Government furnished software.....	48
5.8.1.8	Preparation of quality records.....	48
5.8.1.9	Quality reporting.....	48
5.8.1.10	Corrective action system.....	48
5.8.1.11	Quality cost data.....	49
5.8.2	Products - Software Quality Evaluation.....	49
5.8.2.1	Quality records.....	49
5.8.2.2	Quality reports.....	49
5.8.2.3	Certification.....	49
5.8.3	Independence.....	49
5.9	Software project planning and control.....	49
5.9.1	Activities - Software project planning and control.....	49
5.9.1.1	Sizing and timing assessments.....	49
5.9.1.2	Status and cost reporting.....	50
5.9.1.3	Test documentation control.....	50
5.9.1.4	Software development library (SDL).....	50
5.9.1.5	Risk management.....	50
6.	NOTES.....	51
6.1	Intended use.....	51
6.2	Data requirements list and cross reference.....	51
6.3	Subject term (key word) listing.....	56

CONTENTS - Continued

FIGURES

<u>Figure</u>		<u>Page</u>
1	System development cycle within the system life cycle	2
2	Software development cycle.....	12
3	CSCI sample static structure.....	14
4	System support cycle within the system life cycle.....	62
5	Sequence Construct.....	72
6	IF-THEN-ELSE construct.....	72
7	DO-WHILE construct.....	73
8	DO-UNTIL construct.....	73
9	CASE construct.....	74
10	Relationship among management documents.....	84
11	Relationship among requirements documents.....	85
12	Relationship among design documents.....	87

TABLES

<u>Table</u>		
I	Typical data item requirements.....	82

APPENDIXES

<u>Appendix</u>		
A	List of acronyms and abbreviations.....	59
B	System life cycle.....	61
C	Design and coding standards.....	71
D	Guidelines for tailoring this standard.....	77

1. SCOPE

1.1 Purpose. This standard establishes requirements to be applied during the development and acquisition of Mission-Critical Computer System (MCCS) software, as defined in DOD Directive 5000.29. This standard may also be applied to non-MCCS software development and acquisition.

1.2 Application. Software development is usually an iterative process, in which an iteration of the software development cycle occurs one or more times during each of the system life cycle phases (Figure 1). Appendix B describes a typical system life cycle, the activities that take place during each iteration of software development, and the documentation which typically exists at the beginning of an iteration in any given system life cycle phase. The requirements of this standard shall be applied to each iteration, as described below. The requirements of this standard shall also be applied to the development of software for firmware devices as described in 4.7.

1.2.1 Application to various types of software. This standard applies to deliverable software designated as Computer Software Configuration Items (CSCIs). This standard, or portions thereof, such as configuration management, quality evaluation, and documentation also applies to:

- a. Software developed and delivered as part of a system or a Hardware Configuration Item (HWCI) but not explicitly identified as a CSCI.
- b. Non-deliverable software used in the development and testing of deliverable software and hardware (such as design and test tools).
- c. Deliverable unmodified commercially available and reusable software.
- d. Commercially available software, Government furnished software (GFS), and reusable software that is modified and delivered as part of the system.

The specific requirements of this standard which apply to the above categories will be identified in the statement of work (SOW).

1.2.2 Non-applicability of this standard. This standard, or portions thereof, may not apply to small applications which perform a fixed function that is not expected to change for the life of the system. Guidelines for applying specific portions of this standard to particular categories of software may be found in Appendix D. The SOW will specify the applicable requirements of this standard.

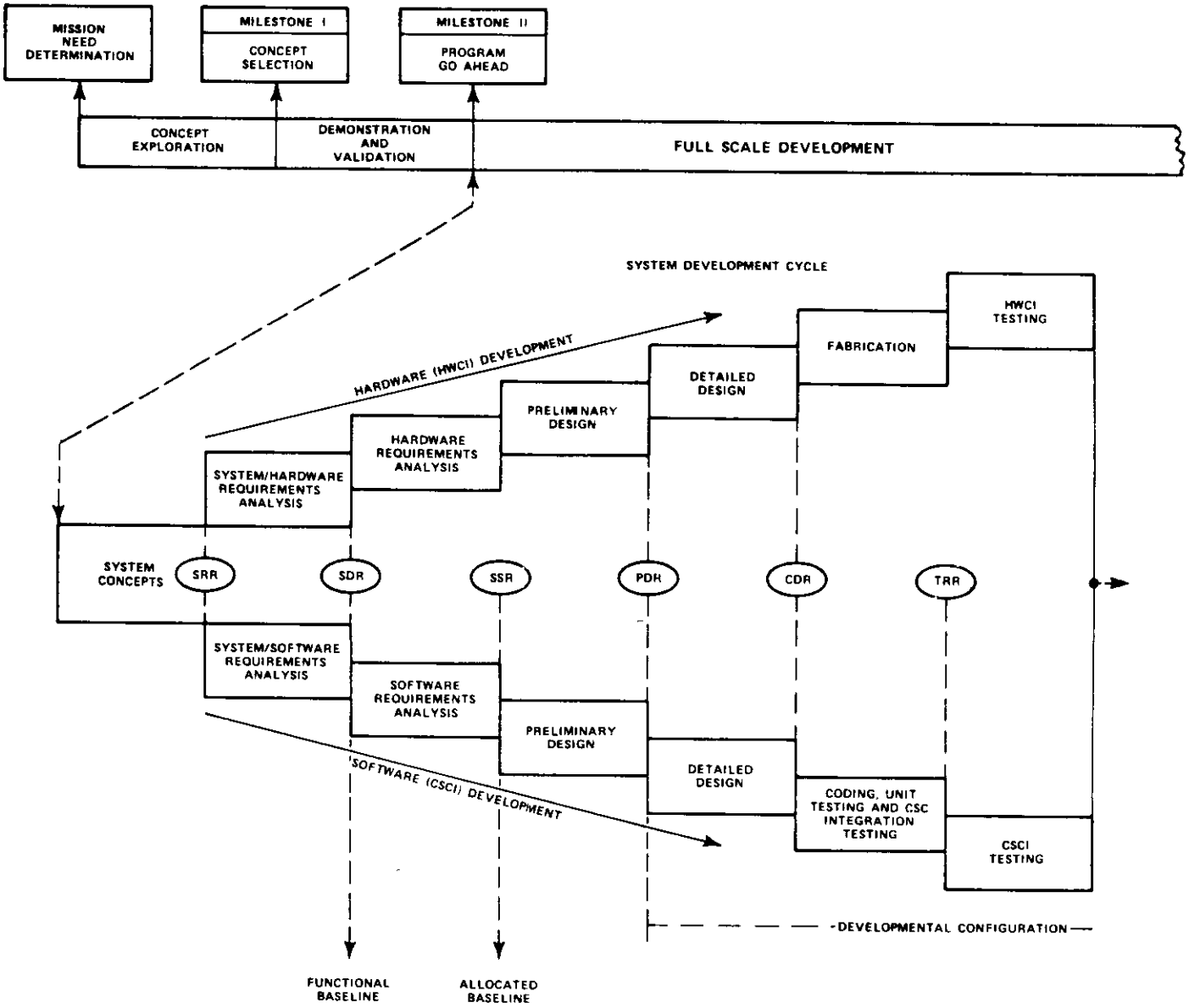


FIGURE 1. System development cycle within the system life cycle.

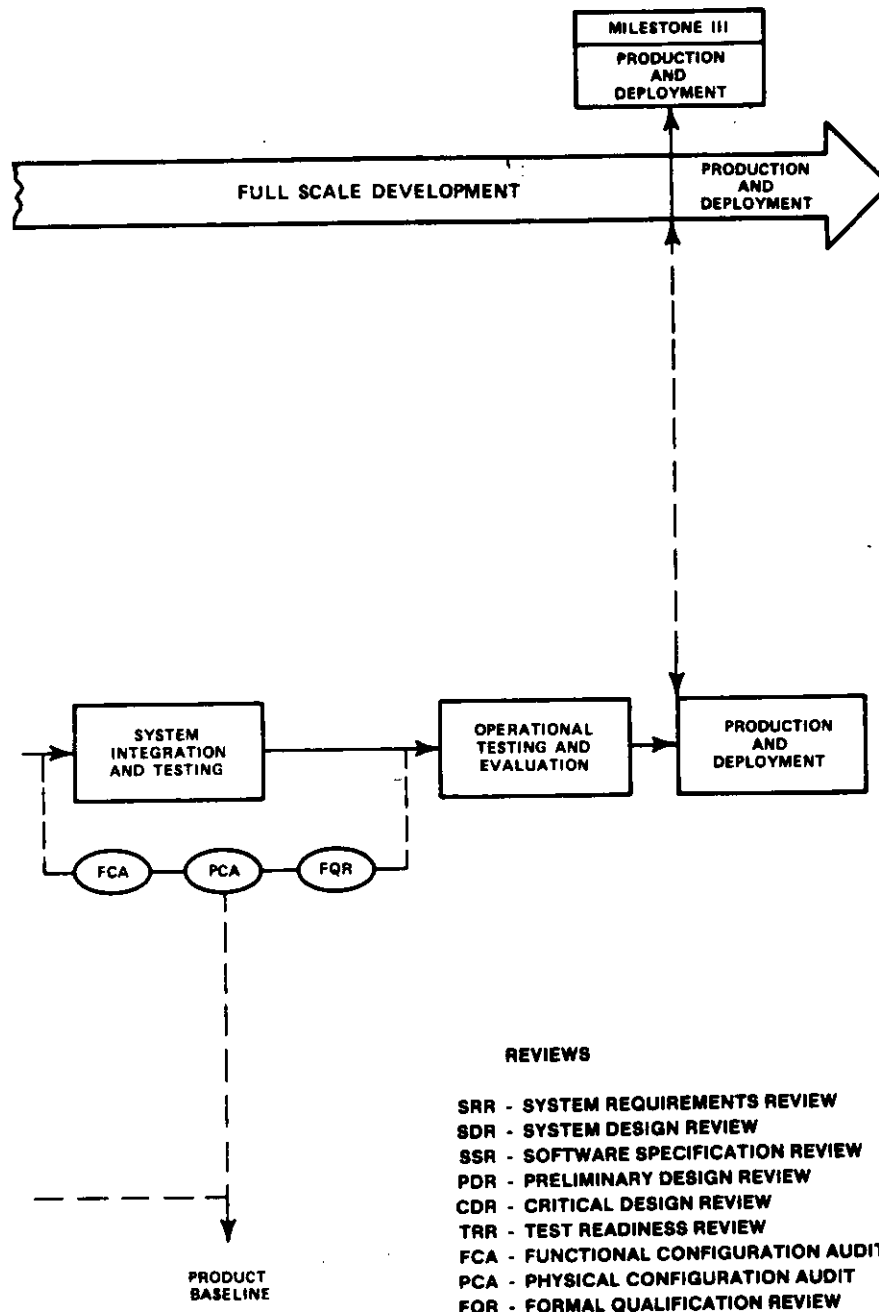


FIGURE 1. System development cycle within the system life cycle. (continued)

1.2.3 Software developed by Government agencies. Although this standard describes software development as performed by a contractor, the provisions of this standard also apply to Government agencies acting as software developers. In this case, the term "contractor" refers to the Government agency that is developing the software. Any contractor of that Government agency is classified as a subcontractor.

1.3 Tailoring of this standard. Software shall be developed in accordance with this standard to the extent specified in the contract clauses, SOW, and the Contract Data Requirements List. Guidelines for applying this standard are provided in Appendix D. The contracting agency will tailor this standard to require only what is needed for each individual acquisition.

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.

STANDARDS

MILITARY

- 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-881 - Work Breakdown Structures for Defense Materiel Items
- MIL-STD-1521 - Technical Reviews and Audits for Systems, Equipments, and Computer Software
- MIL-STD-1535 - Supplier Quality Assurance Program Requirements

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. The initial approved allocated configuration identification as specified in DOD-STD-480.

3.2 Authentication. The procedure (essentially approval) used by the Government in verifying that specification content is acceptable. Authentication does not imply acceptance or responsibility by the Government for the specified item to perform successfully.

3.3 Baseline. A configuration identification document or a set of such documents (regardless of media) formally designated and fixed at a specific time during a configuration item's life cycle. Baselines, plus approved changes from those baselines, constitute the current configuration identification.

3.4 Certification. A process, which may be incremental, by which a contractor provides evidence to the contracting agency that a product meets contractual or otherwise specified requirements.

3.5 Computer data definition. A statement of the characteristics of 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.

3.6 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.

3.7 Computer Software Component (CSC). A functional or logically distinct part of a computer software configuration item. Computer software components may be top-level or lower-level.

3.8 Computer Software Configuration Item (CSCI). See Configuration Item.

3.9 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.

3.10 Computer software quality (or software quality). The degree to which the attributes of the software enable it to perform its specified end item use.

3.11 Configuration Identification. The current approved or conditionally approved technical documentation for a configuration item as set forth in specifications, drawings, and associated lists, and documents referenced therein.

3.12 Configuration Item. Hardware or software, or an aggregation of both, which is designated by the contracting agency for configuration management.

3.13 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 configuration of the design, coding, and testing effort. Any item in the Developmental Configuration may be stored on electronic media.

3.14 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. The definition also applies to read-only digital data that may be used by electronic devices other than digital computers.

3.15 Formal test. A test conducted in accordance with test plans and procedures approved by the contracting agency and witnessed by an authorized contracting agency representative, to show that the software satisfies a specified requirement.

3.16 Functional Baseline. The initial approved functional configuration identification as specified in DOD-STD-480.

3.17 Hardware Configuration Item (HWCI). See Configuration Item.

3.18 Informal test. Any test which does not meet all the requirements of a formal test.

3.19 Modular. Pertaining to software that is organized into limited aggregates of data and contiguous code that perform identifiable functions.

3.20 Product Baseline. The initial approved product configuration identification as specified in DOD-STD-480.

3.21 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. A software development library provides storage of and controlled access to software and documentation in both human-readable and machine-readable form. The library may also contain management data pertinent to the software development project.

3.22 Top-down. Pertaining to an approach that starts with the highest level of a hierarchy and proceeds through progressively lower levels. For example, top-down design, top-down coding, top-down testing.

3.23 Unit. The smallest logical entity specified in the detailed design which completely describes a single function in sufficient detail to allow implementing code to be produced and tested independently of other Units. Units are the actual physical entities implemented in code.

4. GENERAL REQUIREMENTS

4.1 Software development cycle. The contractor shall implement a software development cycle that includes the following six phases:

- a. Software Requirements Analysis
- b. Preliminary Design
- c. Detailed Design
- d. Coding and Unit Testing
- e. Computer Software Component (CSC) Integration and Testing
- f. CSCI Testing.

4.1.1 Each iteration of the software development cycle, regardless of the system life cycle phase during which it occurs, is initiated by allocation of system requirements to that software or a subsequent revision to those requirements.

4.1.2 The relationship of the software development cycle phases with the products, reviews and audits, and baselines and Developmental Configurations required by Section 5 of this Standard are shown in Figure 2. Figure 2 reflects the sequential phases of a software development cycle, as well as the documentation which typically exists prior to initiating an iteration. During software development, more than one iteration of the software development cycle may be in progress at the same time. Each iteration represents a different version of the software. This process may be described as an "evolutionary acquisition" or "incremental build" approach. Within each iteration, the software development phases also typically overlap, rather than form a discrete termination-initiation sequence. For example, performing Unit code and test concurrently with CSC integration and test is useful in implementing incremental builds. The relationship of the software development cycle to the system life cycle, including system allocation of requirements to CSCIs, and system integration and testing of HWCIs and CSCIs, is described in Appendix B.

4.2 Computer software organization. Computer software developed in accordance with this standard shall be organized as one or more CSCIs or other types of software (see 1.2.1). Each CSCI is part of a system, segment, or prime item and shall consist of one or more Top Level Computer Software Components (TLCSCs). Each TLCSC shall consist of Lower-Level Computer Software Components (LLCSCs) or Units. LLCSCs may consist of other LLCSCs or Units. TLCSCs and LLCSCs are logical groupings. Units are the smallest logical entities, and the actual physical entities implemented in code. The static structure of CSCIs, TLCSCs, LLCSCs, and Units shall form a hierarchical structure as illustrated in Figure 3. The hierarchical structure shall uniquely identify all CSCIs, TLCSCs, LLCSCs, and Units.

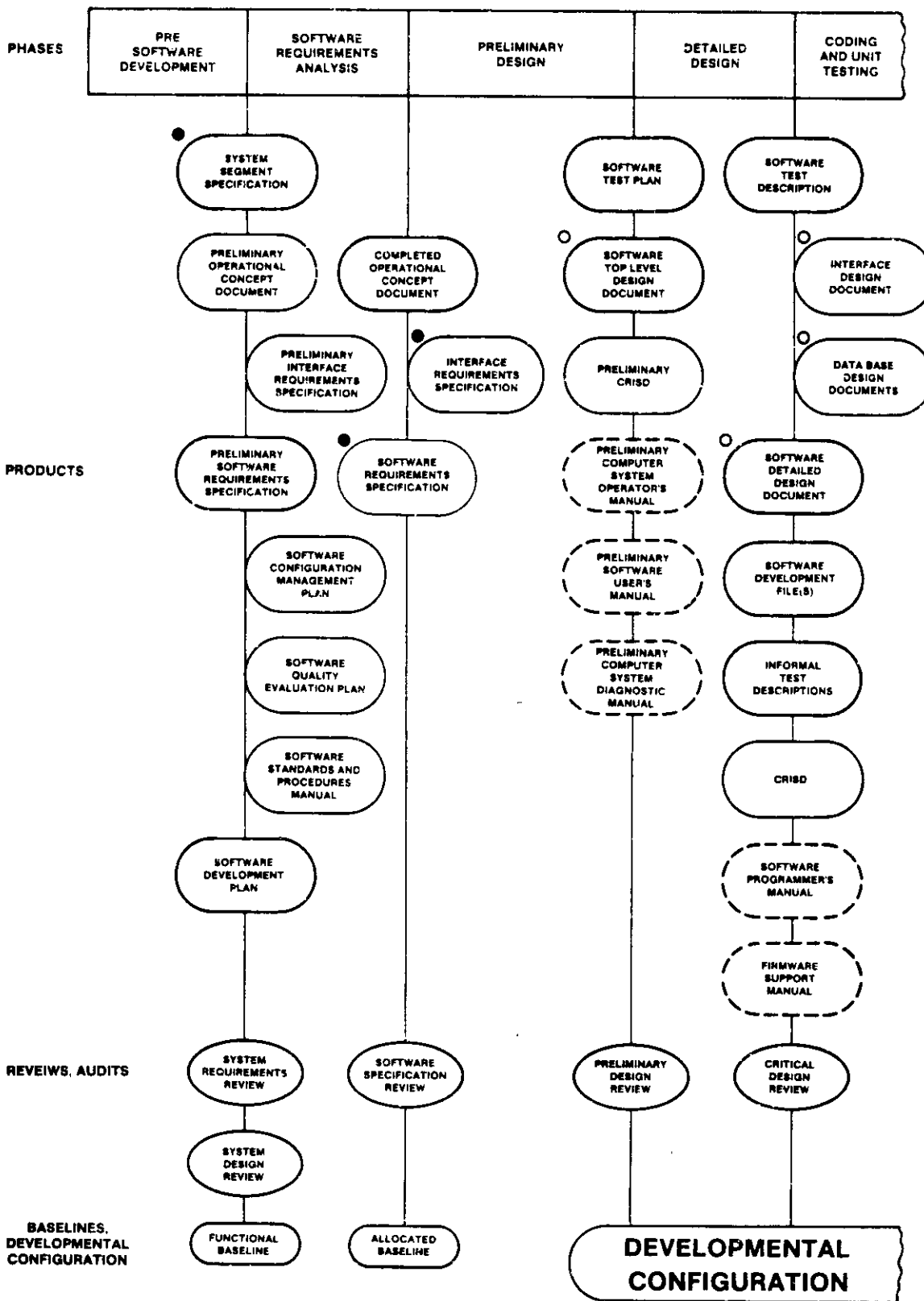


FIGURE 2. Software development cycle.

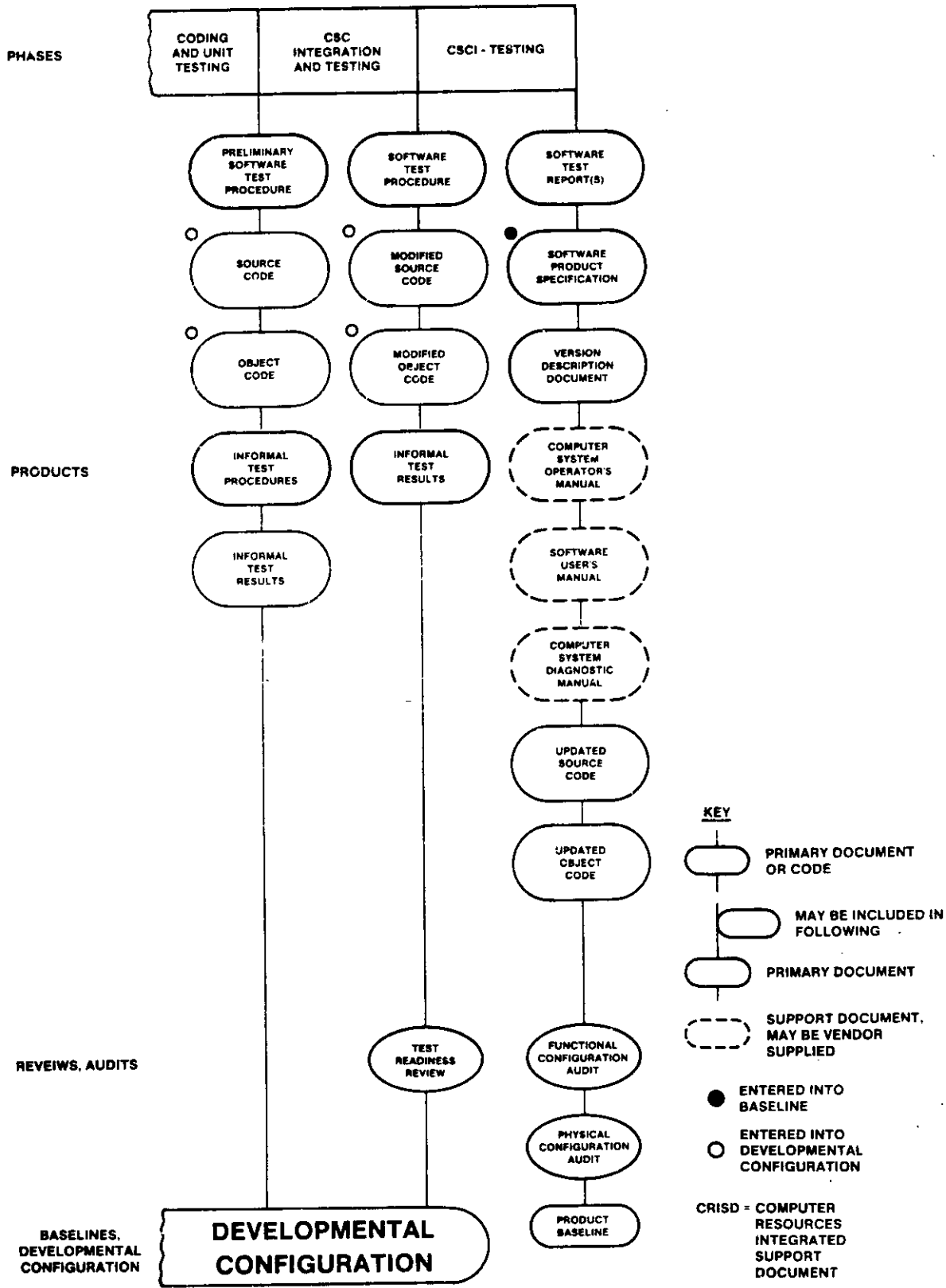


FIGURE 2. Software development cycle. (continued)

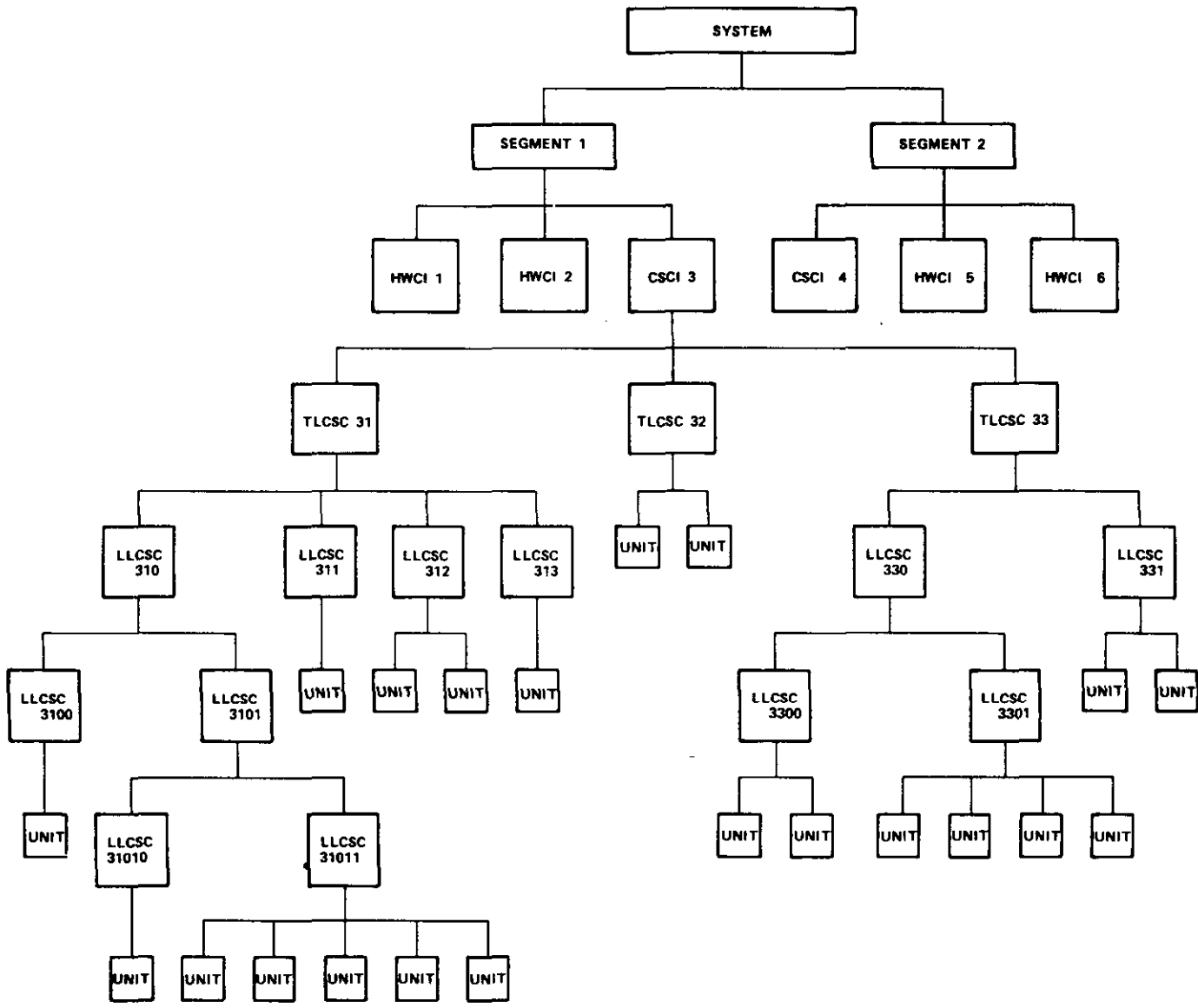


FIGURE 3. CSCI sample static structure.

4.2.1 The partitioning of the CSCI into TLCSCs, LLCSCs, and Units may be based on functional requirements, data flow requirements, or other design considerations. The hierarchical structure illustrated in Figure 3 demonstrates the static relationship of the TLCSCs, LLCSCs, and Units based on the partitioning considerations and does not represent either the control flow of the software during execution or the implemented code. Guidelines for selecting CSCIs are contained in MIL-STD-483, Appendix XVII. These guidelines may also be applied to selecting TLCSCs, LLCSCs, and Units.

4.3 Software quality. The contractor shall plan and implement the software development project with the objective of building in quality. To achieve this quality, the contractor shall:

- a. Establish and maintain a complete set of requirements for the software. These requirements shall serve as the standard against which software quality is evaluated. To establish the requirements, the contractor shall perform the tasks specified in 5.1. To maintain the requirements, the contractor shall perform the tasks specified in 5.7.
- b. Establish and implement a complete process, including methodologies and tools, for developing the software and its documentation. The process shall be designed to build quality into the software and its documentation and to maintain the level of quality throughout the life cycle of the software. The development process shall include both contractor internal steps (specified in the Software Development Plan (SDP), Software Configuration Management Plan (SCMP), and Software Standards and Procedures Manual (SSPM)), and the formal steps specified in 5.1 through 5.7, and 5.9 (see 6.2).
- c. Establish and maintain a process to evaluate the software, associated documentation, and the software development process. The objective of this process shall be to improve the quality of the software and its documentation, by providing feedback and ensuring that necessary corrections are made. The quality evaluation process shall include both contractor internal steps (specified in either the SDP or the Software Quality Evaluation Plan (SQEP)) and the formal steps specified in 5.8 (see 6.2).

4.4 Use of commercially available, reusable, and Government furnished software. In order to facilitate cost-effective development and support of MCCS software, the contractor is encouraged to incorporate into the current software design commercially available software, GFS, and reusable software developed for other applications. However, the contractor shall perform the following activities prior to incorporating commercially available software, reusable software, GFS, or any combination of these, into the design: (1) describe in the SDP

the data rights and documentation the contractor plans to provide the contracting agency regarding the commercially available and reusable software, (2) evaluate the commercially available, reusable, or Government furnished software to determine whether it performs as documented, (3) describe in the SDP the contractor's plans for certifying the commercially available or reusable software, and (4) obtain explicit contracting agency approval for use of commercially available software (see 5.8.1.7 and 6.2).

4.5 Subcontractor control. The contractor shall ensure that all subcontractors developing software and documentation comply with subcontract requirements. The requirements of 4.4 shall apply to commercially available and reusable software procured from subcontractors. Additional guidance may be found in MIL-STD-1535.

4.6 Non-deliverable software, firmware, and hardware. The contractor shall describe in the SDP the controls to be imposed on all non-deliverable software, firmware, and hardware used in the development and acquisition of deliverable software (see 6.2). As a minimum, the contractor shall describe the provisions for:

- a. Modifications (as applicable)
- b. Documentation
- c. Configuration Management
- d. Design & Coding Standards
- e. Testing
- f. Quality Evaluation
- g. Certification.

4.7 Firmware. The application of the requirements in this standard to firmware depends on whether the firmware is designated as a CSCI or as part of an HWCI. If the software to be implemented in firmware is designated as a CSCI, all the requirements in this standard apply, as tailored by the contract. If the software to be implemented in firmware is considered part of an HWCI, the contractor shall identify the applicable requirements in the SDP and apply these requirements subject to contracting agency approval (see 6.2).

4.8 Development methodologies. The contractor shall use a top-down approach to design, code, integrate, and test all CSCIs, unless specific alternate methodologies have been proposed in either the SSPM or SDP (see Appendix D) and received contracting agency approval (see 6.2).

4.9 Security. The contractor shall implement applicable security measures during software design and development.

4.10 Deliverable Data. Deliverable data prepared in accordance with the requirements of sections 4 and 5 of this standard and identified on the DD Form 1423, Contract Data Requirements List, shall be prepared in accordance with the instructions in the applicable DIDs (see 6.2).

4.11 Deviations and waivers. The contractor and, if applicable, subcontractors shall develop software in compliance with this standard, as required by the contract, unless a deviation or waiver has been approved by the contracting agency in accordance with DOD-STD-480 or MIL-STD-481.

5. DETAILED REQUIREMENTS

5.1 Software Requirements Analysis. The contractor shall define and analyze a complete set of functional, performance, interface, and qualification requirements for each CSCI. These requirements shall be derived from the system requirements as defined in the System/Segment Specification (SSS), prime item specification, critical item specification, or other sources specified in the contract. Additional requirements may be derived during the analysis and allocation of system-derived requirements. The contractor shall also prepare or update, as applicable (see Appendix B), an SDP, SSPM, SCMP, SQEP, and Operational Concept Document (OCD), and establish internal control over these documents.

5.1.1 Activities - Software Requirements Analysis. The contractor shall perform the following activities during Software Requirements Analysis.

5.1.1.1 If plans for developing the software have not previously been prepared and approved by the contracting agency (see Appendix B), the contractor shall prepare them. The contractor's plans for software development shall include:

- a. Resources and organization, describing: (1) the contractor's facilities, (2) Government furnished equipment, software, and services required, and (3) organizational structure, personnel, and resources for software development, software configuration management, and software quality evaluation.
- b. Development schedule and milestones, describing: (1) each individual activity of the project, (2) the activity network, (3) risk management procedures, and (4) identifiable high risk areas.
- c. Software standards and procedures, describing: (1) tools, techniques, and methodologies to be used in the development, (2) if applicable, criteria for departing from a top-down approach (see 5.3.1.3, 5.3.1.4), (3) the software development library and associated access and control procedures, (4) the format and contents of software development files, associated procedures, and organizational responsibilities, (5) the format and contents of all informal test documentation, (6) design and coding standards, and (7) procedures and reports used to prepare for formal reviews.
- d. Software configuration management, describing: (1) configuration identification procedures, (2) configuration control including software problem and change reports, and

review boards, (3) configuration status accounting, (4) configuration audits, (5) preparation for configuration authentication procedures, and (6) configuration management major milestones.

- e. Software quality evaluation, describing: (1) evaluation of development plans, standards, and procedures, (2) evaluation of the contractor's compliance with those plans, standards, and procedures, (3) evaluation of the products of software development, (4) implementation of a quality evaluation reporting system, and (5) implementation of a corrective action system.
- f. Commercially available, reusable, and Government furnished software, describing: (1) rationale for its use, (2) plans for providing associated data rights and documentation for commercially available and reusable software, (3) plans for determining that the commercially available, reusable, and Government furnished software performs as documented, and (4) plans for certifying commercially available and reusable software.
- g. Data rights and documentation for the software development library (SDL), describing the plans for providing associated data rights and documentation for the SDL.
- h. Subcontractor control, describing: (1) the organization responsible for subcontractor control, and (2) the procedures to ensure that all subcontractor-developed software meets subcontract requirements.
- i. Control provisions for non-deliverable software, firmware, and hardware, describing requirements for: (1) modifications (if applicable), (2) documentation, (3) configuration management, (4) design and coding standards, (5) testing, (6) quality evaluation, and (7) certification.
- j. Control provisions for software that is part of a hardware item, describing procedures for: (1) requirements analysis and allocation, (2) design and coding, (3) hardware and software integration and test, (4) coordination of hardware and software design, (5) documentation, (6) software configuration management, and (7) software quality evaluation.
- k. Interface management with associate contractors, describing the contractor's plan for coordinating development and data management efforts to ensure interface compatibility.

5.1.1.2 The contractor shall establish internal control over the SDP, SSPM, SCMP, and SQEP. The contractor shall monitor the development effort for consistency with the SDP, SSPM, SCMP, and SQEP (see 5.8.1.2.2). The contractor shall notify the contracting

agency of proposed changes to these documents and make necessary revisions. All proposed changes shall be subject to disapproval by the contracting agency. In addition, the contractor shall notify the contracting agency at the next review, audit, or in the next status report (whichever comes first) of any actions or procedures occurring during Software Requirements Analysis that deviate from the SDP, SSPM, SCMP, or SQEP.

5.1.1.3 If provided by the Government, the contractor shall analyze the preliminary OCD (see Appendix B) for adequacy, understandability, validity, and completeness.

5.1.1.4 The contractor shall identify and describe the mission of the system and its operational and support environments. The contractor shall also describe the functions and characteristics of the computer system within the overall system (see 5.1.2.2).

5.1.1.5 The contractor shall analyze the SSS and, if provided, the preliminary Software Requirements Specifications (SRSS) and Interface Requirements Specifications (IRSS) for adequacy, testability, understandability, validity, and completeness. Circumstances under which the specifications are provided by the Government are described in Appendix B.

5.1.1.6 The contractor shall define a complete set of functional, performance, interface, and qualification requirements for each CSCI, incorporating the results of 5.1.1.5. Requirements specified by the contractor shall also include the following areas:

- a. Programming constraints and standards
- b. Design constraints and standards
- c. Adaptation
- d. Quality factors
- e. Preparation for delivery.

5.1.1.7 In the definition and analysis of software requirements, the contractor shall use structured requirements analysis tools, techniques, or a combination of both. The specific tools and techniques to be used shall be identified in either the SSPM or SDP (see Appendix D) and shall be subject to contracting agency approval. The contractor shall map the requirements defined in 5.1.1.6 to the applicable higher-level documents.

5.1.1.8 The contractor shall conduct internal in-process reviews during this phase (see 5.8.1.2.3) and shall make all necessary changes based on the results of the internal reviews prior to presenting the requirements document(s) to the contracting agency.

5.1.2 Products - Software Requirements Analysis. The contractor shall produce the following products during Software Requirements Analysis (see 6.2).

5.1.2.1 The contractor shall prepare or produce updated versions of (whichever is applicable, see Appendix B) the SDP, SSPM, SCMP, and SQEP.

5.1.2.2 The contractor shall produce an OCD for the system. In the event a preliminary OCD has been provided by the Government, the contractor shall update and complete the document.

5.1.2.3 The contractor shall produce records and summary reports of the internal reviews conducted (see 5.8.2.1 and 5.8.2.2).

5.1.2.4 The contractor shall produce an SRS and, if applicable, IRS(s) for each CSCI (see Appendix D). In the event preliminary SRSs and IRSs have been provided by the Government, the contractor shall produce updated and completed versions of these specifications (see Appendix B). Additional guidance on preparing specifications is provided in MIL-STD-490.

5.1.3 Formal Reviews - Software Requirements Analysis. The contractor shall present the newly prepared or updated OCD for the system and an SRS and IRS(s) for each CSCI at a Software Specification Review (SSR). The purpose of the SSR is to demonstrate to the contracting agency the adequacy of the OCD, SRS, and, if applicable, IRS(s). Specific details regarding the SSR process are contained in MIL-STD-1521.

5.1.4 Baselines - Software Requirements Analysis. Upon completion of the SSR, and when authenticated by the contracting agency, the SRS and IRS(s) will establish the Allocated Baseline for each CSCI. Specific details regarding the baseline process are contained in MIL-STD-483 and MIL-STD-490.

5.2 Preliminary Design. The contractor shall develop a top-level design of each CSCI which completely reflects the requirements specified in the SRS and IRS(s). The contractor may develop lower-level design for critical elements of the CSCI. The criteria for determining critical elements shall be described in either the SSPM or SDP (see Appendix D).

5.2.1 Activities - Preliminary Design. The contractor shall perform the following activities during Preliminary Design.

5.2.1.1 The contractor shall monitor the development effort for consistency with the SDP, SSPM, SCMP, and SQEP (see 5.8.1.2.2). The contractor shall notify the contracting agency of proposed changes to these documents, and make necessary revisions. All proposed changes shall be subject to disapproval by the contracting agency. In addition, the contractor shall notify the contracting agency at the next review, audit, or in the next

status report (whichever comes first) of any actions or procedures occurring during Preliminary Design that deviate from the SDP, SSPM, SCMP, or SQEP.

5.2.1.2 The contractor shall establish the top-level design of each CSCI by allocating requirements from the SRS and, if applicable, IRS(s) to the TLCSCs of each CSCI. In defining each TLCSC the contractor shall identify:

- a. The TLCSC's place in the CSCI's static structure
- b. Functions allocated to the TLCSC
- c. Memory size and processing time allocated (including reserve capacities) to the TLCSC
- d. Functional control and data flow to and from the TLCSC
- e. Known interrupt and special control features (such as non-standard subroutine returns) of the TLCSC
- f. Global data shared with other TLCSCs
- g. Applicable inputs, local data, interrupts, timing and sequencing, processing, and outputs of the TLCSC
- h. Adaptation data needed by the TLCSC.

5.2.1.3 The contractor may establish the lower-level design of critical elements of each CSCI, including external interfaces and data bases, by refining TLCSCs to LLCSCs and Units. The criteria for determining critical elements shall be described in either the SSPM or SDP (see Appendix D).

5.2.1.4 In establishing and defining the top-level and, as applicable, lower-level design of each CSCI, the contractor shall use a program design language or some other top-level design description tool or methodology. This tool or methodology shall be identified in either the SSPM or SDP (see Appendix D) and shall be subject to contracting agency approval.

5.2.1.5 In the development of the top-level design, the contractor shall incorporate applicable human factors engineering principles, including:

- a. Human information processing capabilities and limitations
- b. Anthropometric characteristics of the target population
- c. Foreseeable human errors under both normal and extreme conditions
- d. Implications for the total system environment (to include

training, support, maintenance, and operational environment).

5.2.1.6 The contractor shall develop test plans for both informal and formal tests.

- a. Informal tests shall test individual Units during Coding and Unit Testing and aggregates of Units during CSC Integration and Testing. For Unit testing, the contractor shall identify the overall test requirements, test responsibilities, and schedule information. For CSC integration testing, the contractor shall identify: (1) the overall test requirements, test responsibilities, and schedule information, and (2) different classes of CSC integration tests. Although informal test documentation does not require Government approval, it shall be made available for Government review.
- b. Formal tests shall test the fully implemented CSCI during CSCI Testing, to show that the CSCI satisfies its specified requirements. Formal tests may also occur at the TLCSC, LLCSC, and Unit levels, when compliance with specified requirements cannot be shown at the CSCI level. Some CSCIs may require integration with other computer systems, HWCIs, or CSCIs before all formal testing can be completed. For formal testing, the contractor shall identify: (1) the test requirements applicable to CSCI testing, (2) CSCI test organization, responsibilities, and schedule information, (3) different classes of formal tests, (4) data recording, reduction, and analysis requirements, and (5) the purpose of each formal test planned. The contractor shall plan for documenting formal test results as well. All individuals responsible for planning formal tests shall be sufficiently independent from the individuals responsible for development to permit objective testing.
- c. The contractor shall identify all the resources (facilities, personnel, hardware, software) required for informal and formal testing.

5.2.1.7 The contractor shall define a preliminary version of the procedures and information for the operation of the computer system in which each CSCI executes (see 5.2.2.6). This definition shall include:

- a. System preparation and set up
- b. Operating procedures
- c. Input/Output
- d. Monitoring procedures
- e. Off-line routines
- f. Recovery and special procedures
- g. Diagnostic features.

5.2.1.8 The contractor shall define a preliminary version of the instructions for user personnel to execute each CSCI requiring user interaction (see 5.2.2.6). This definition shall include for each function the CSCI performs:

- a. Name, number and purpose of the function
- b. Initialization requirements
- c. Execution options
- d. User and system inputs
- e. Termination and restart procedures
- f. Expected outputs
- g. Interrelationship with other functions
- h. Error messages
- i. Diagnostic features.

5.2.1.9 The contractor shall define a preliminary version of the information necessary to identify a computer system malfunction and instructions to run the diagnostics (see 5.2.2.6). This definition shall include:

- a. Identification of all support hardware, software, and procedures necessary to perform system diagnosis.
- b. A description of each diagnostic tool available for the system.
- c. A description of each diagnostic test available on the diagnostic tools, including: (1) the purpose of each test, (2) procedures for executing the test, (3) additional hardware, software, or firmware necessary for executing the test, and (4) all diagnostic messages.

5.2.1.10 The contractor shall define a preliminary version of the information required to perform life cycle support for the contractually deliverable software (see 5.2.2.6). This definition shall include identification of:

- a. The support environment, describing required: (1) support software, (2) equipment, (3) facilities, and (4) personnel.
- b. Support operations, describing: (1) general usage instructions (initiation, general operation, and monitoring operations of the support environment), (2) administration, (3) software modification, (4) software integration and testing, (5) system and software generation, (6) software quality evaluation, (7) corrective action system, (8) configuration management, (9) simulation, (10) emulation, (11) reproduction, and (12) operational distributions.
- c. Training plans and provisions.
- d. Predicted level of change to the deliverable software in the support environment.

5.2.1.11 The contractor shall conduct internal in-process reviews during this phase (See 5.8.1.2.4) and shall make all necessary changes based on the results of the internal reviews, prior to presenting the top-level design, test plans, and operation and support documents to the contracting agency.

5.2.2 Products - Preliminary Design. The contractor shall produce the following products during Preliminary Design (see 6.2).

5.2.2.1 The contractor shall produce updated versions of the SDP, SSPM, SCMP, and SQEP as necessary.

5.2.2.2 The contractor shall produce records and summary reports of the internal reviews conducted (see 5.8.2.1 and 5.8.2.2).

5.2.2.3 The contractor shall produce a Software Top-Level Design Document (STLDD) for each CSCI to describe the top-level design of the CSCI.

5.2.2.4 The contractor may produce preliminary versions of the Software Detailed Design Document (SDDD), Interface Design Document(s) (IDD(s)), and Data Base Design Document(s) (DBDD(s)) for critical lower-level elements of the CSCI.

5.2.2.5 The contractor shall produce a Software Test Plan (STP) for each CSCI to describe the plans for both informal and formal testing of the CSCI.

5.2.2.6 The contractor shall produce preliminary versions of the:

- a. Computer System Operator's Manual (CSOM)
- b. Software User's Manual (SUM) for one or more CSCIs
- c. Computer System Diagnostic Manual (CSDM)
- d. Computer Resources Integrated Support Document (CRISD).

5.2.3 Formal Reviews - Preliminary Design. The contractor shall present the STLDD and the STP for each CSCI, and preliminary versions of the CSOM, SUM(s), CSDM, and CRISD at a Preliminary Design Review (PDR). The purpose of the PDR is to review the top-level design, test plans, and preliminary operation and support documents with the contracting agency and to demonstrate to the contracting agency that: (1) the top-level design satisfies the software requirements allocated from the higher-level documents, (2) the test plans establish adequate test criteria for each CSCI and address all specified requirements, and (3) the preliminary versions of the CSOM, SUM(s), CSDM, and CRISD will, in final form, adequately address the operation and support of the computer system. In addition, the PDR may review preliminary versions of the SDDD, IDD(s), and DBDD(s) for critical lower-level elements, including external interfaces and data

base(s), to demonstrate that the lower-level design for critical elements will satisfy the specified requirements. Specific details regarding the PDR process are contained in MIL-STD-1521.

5.2.4 Developmental Configuration - Preliminary Design. Upon successful completion of the PDR, the STLDD shall establish the contractor's Developmental Configuration for each CSCI.

5.3 Detailed Design. The contractor shall develop a modular, detailed design for each CSCI.

5.3.1 Activities - Detailed Design. The contractor shall perform the following activities during Detailed Design.

5.3.1.1 The contractor shall monitor the development effort for consistency with the SDP, SSPM, SCMP, and SQEP (see 5.8.1.2.2). The contractor shall notify the contracting agency of proposed changes to these documents and make necessary revisions. All proposed changes shall be subject to disapproval by the contracting agency. In addition, the contractor shall notify the contracting agency at the next review, audit, or in the next status report (whichever comes first) of any actions or procedures occurring during Detailed Design that deviate from the SDP, SSPM, SCMP, or SQEP.

5.3.1.2 The contractor shall establish the complete, modular, lower-level design for each CSCI, by refining TLCSCs into LLCSCs and Units. Each Unit shall perform a single function. In refining TLCSCs, the contractor shall identify:

- a. All required details for implementing external interfaces, including item summary and item format for each interface
- b. Global data definitions within each TLCSC
- c. Inputs, local data definitions, process control requirements, processing, utilization of other elements, limitations, and outputs of all LLCSCs
- d. Inputs, local data definitions, process control requirements, processing, special control features, protection, error handling, utilization of other elements, limitations, and outputs for all Units
- e. Detailed data base design including data base management system overview, data base structure, data base file design, and data base references.

5.3.1.3 The contractor shall refine all TLCSCs using a top-down design approach, unless specific alternate methodologies have been proposed in either the SSPM or SDP (see Appendix D) and have received contracting agency approval.

5.3.1.4 The contractor may depart from a top-down approach to: (1) address critical lower-level elements or (2) incorporate commercially available, reusable, and Government furnished software. The contractor shall describe the criteria for determining critical lower-level elements in either the SSPM or SDP (see Appendix D). Examples of criteria for determining criticality are software performance, cost, and schedule.

5.3.1.5 In the development of the detailed design for each CSCI, the contractor shall employ a program design language. The language and other tools to be used shall be identified in either the SSPM or SDP (see Appendix D) and shall be subject to contracting agency approval.

5.3.1.6 The contractor shall ensure that the detailed design incorporates applicable human factors engineering principles (see 5.2.1.5).

5.3.1.7 The contractor shall monitor size and time estimates for the CSCI and adjust the estimates, if necessary. All modifications to controlled or baselined documentation shall be made in accordance with the configuration management requirements contained herein (see 5.7).

5.3.1.8 The contractor shall establish software development files (SDFs) for all Units. Each SDF may serve a single Unit or logically related group of Units. Unit requirements, design considerations and constraints, schedule, status information, and test documentation shall be incorporated into the corresponding SDF. All SDFs shall be in the format described in either the SSPM or SDP (see Appendix D). To reduce duplication, SDFs should not contain information provided in other documents. SDFs may be generated, maintained, and controlled by automated means.

5.3.1.9 The contractor shall document additional engineering information generated in the design process for each CSCI. The engineering information shall include rationale, results of analyses and trade-off studies, and any other information which aids in understanding the detailed design.

5.3.1.10 The contractor shall identify the test requirements, responsibilities, and schedule for the informal testing to be conducted for each Unit, and shall record them in the corresponding SDF.

5.3.1.11 The contractor shall describe test cases for each informal Unit test in terms of inputs, expected results, and evaluation criteria. The test cases for each Unit shall be described in the corresponding SDF.

5.3.1.12 The contractor shall identify the requirements, responsibilities, and schedule for each CSC integration test.

5.3.1.13 The contractor shall describe test cases for each informal CSC integration test in terms of inputs, expected results, and evaluation criteria.

5.3.1.14 The contractor shall describe test cases for each formal CSCI test identified in the STP. Test case descriptions shall include:

- a. Initialization requirements
- b. Input data
- c. Expected intermediate test results
- d. Expected output data
- e. Criteria for evaluating results
- f. Assumptions and constraints.

5.3.1.15 The contractor shall update with any additional known details all information and instructions pertaining to computer system operation, software operation by users, and computer system diagnostics (see 5.3.2.9).

5.3.1.16 The contractor shall complete the information that is required to perform life cycle support of the contractually deliverable software (see 5.3.2.10).

5.3.1.17 The contractor shall prepare information to facilitate programming or reprogramming software for the target computer (see 5.3.2.11). The information shall include:

- a. Equipment configuration
- b. Operational characteristics, capabilities, and limitations
- c. Compilation and assembly information
- d. Programming features
- e. Program instructions
- f. I/O control features
- g. Examples of programming techniques
- h. Special features
- i. Error detection and diagnostic features.

5.3.1.18 The contractor shall describe the information necessary to modify or replace the read-only memory (ROM), programmable read-only memory (PROM), and other such firmware components of the system (see 5.3.2.11). This description shall include:

- a. Description of firmware components
- b. Installation and repair procedures
- c. Security implications
- d. Operational and environment limitations
- e. Hardware needed for programming firmware devices

- f. Software needed for programming firmware devices
- g. Procedures for programming firmware devices
- h. Vendor information.

5.3.1.19 The contractor shall conduct internal in-process reviews during this phase (see 5.8.1.2.5) and shall make all necessary changes based on the results of the internal review, prior to presenting the detailed design, formal test case documentation, and operation and support documentation to the contracting agency.

5.3.2 Products - Detailed Design. The contractor shall produce the following products during Detailed Design (see 6.2).

5.3.2.1 The contractor shall produce updated versions of the SDP, SSPM, SCMP, and SQEP as necessary.

5.3.2.2 The contractor shall produce records and summary reports of the internal reviews conducted (see 5.8.2.1 and 5.8.2.2).

5.3.2.3 The contractor shall produce an SDDD for each CSCI, to describe the detailed design. The contractor shall include in the SDDD, in Section 6 Notes, additional engineering information (rationale, results of analyses and trade-off studies, etc.) which aids in understanding the detailed design of the CSCI.

5.3.2.4 The contractor shall produce an IDD for each IRS to describe the details of external interfaces. The contractor shall include in the IDD(s), in Section 6 Notes, additional information (rationale, results of analyses and trade-off studies, etc.) which aids in understanding the details of external interfaces.

5.3.2.5 The contractor shall produce one or more DBDDs. Each DBDD shall describe the contents and structure of one or more data bases. (Data base interactions and control mechanisms are described in the top-level and detailed design documents). The contractor shall include in the DBDD(s), in Section 6 Notes, additional information (rationale, results of analyses and trade-off studies, etc.) which aids in understanding the details of the data base(s).

5.3.2.6 The contractor shall establish and maintain SDFs for all Units.

5.3.2.7 The contractor shall produce documents that identify each informal CSC integration test and describe the test cases, in the standard format described in either the SSPM or SDP (see Appendix D), for each informal test to be executed.

5.3.2.8 The contractor shall produce a Software Test Description (STD) for each CSCI, to define test cases for each formal test of the CSCI described in the STP.

5.3.2.9 The contractor shall produce updated versions of the CSOM,

SUM(s), and CSDM.

5.3.2.10 The contractor shall produce a completed CRISD.

5.3.2.11 The contractor shall produce a Software Programmer's Manual (SPM) and a Firmware Support Manual (FSM).

5.3.3 Formal Reviews - Detailed Design. The contractor shall present the SDDD and the STD for each CSCI at a Critical Design Review (CDR). The contractor shall also present the IDD(s), DBDD(s), SPM, FSM, and updated CSOM, SUM, CSDM, and CRISD at this review. The purpose of the CDR is to review the detailed design, test description, and operation and support documents with the contracting agency, and to demonstrate to the contracting agency that: (1) the detailed design satisfies the requirements of the SRS and the IRS(s), (2) the SDDD, IDD(s), and DBDD(s) further refine the design details of the CSCI in a manner consistent with the STLDD, (3) the STD provides adequate test cases for the formal tests identified in the STP, (4) the updated versions of the CSOM, SUM(s), and CSDM will, in final form, adequately address the operation and support of the computer system, and (5) the SPM, FSM, and CRISD adequately address software programming support, firmware support, and integrated computer resources support. Specific details regarding the CDR process are contained in MIL-STD-1521.

5.3.4 Developmental Configuration - Detailed Design. Upon successful completion of the CDR, the contractor shall enter the SDDD, IDD(s), and the DBDD(s) for each CSCI into the Developmental Configuration for the CSCI.

5.4 Coding and Unit Testing. The contractor shall code and test each Unit making up the detailed design.

5.4.1 Activities - Coding and Unit Testing. The contractor shall perform the following activities during Coding and Unit Testing.

5.4.1.1 The contractor shall monitor the development effort for consistency with the SDP, SSPM, SCMP, and SQEP (see 5.8.1.2.2). The contractor shall notify the contracting agency of proposed changes to these documents, and make necessary revisions. All proposed changes shall be subject to disapproval by the contracting agency. In addition, the contractor shall notify the contracting agency at the next review, audit, or in the next status report (whichever comes first) of any actions or procedures occurring during Coding and Unit Testing that deviate from the SDP, SSPM, SCMP, or SQEP.

5.4.1.2 The contractor shall code and test Units in top-down sequence, unless alternate methodologies have been proposed in either the SSPM or SDP (see Appendix D) and have received contracting agency approval.

5.4.1.3 The contractor may depart from a top-down approach to: (1) code and test critical Units or (2) incorporate commercially available, reusable, or Government furnished software. The contractor shall describe the criteria for determining critical Units in either the SSPM or SDP (see Appendix D). Examples of criteria for determining criticality are software performance, cost, and schedule.

5.4.1.4 The contractor shall code all Units in accordance with coding standards. If the contractor has not proposed use of internal coding standards in either the SSPM or SDP (see Appendix D) and received contracting agency approval for the internal coding standards, then the coding standards of Appendix C shall apply.

5.4.1.5 The contractor shall produce deliverable code that can be regenerated and maintained using only Government-owned, contractually deliverable, or commercially available support software and hardware.

5.4.1.6 Prior to the testing of each Unit, the contractor shall prepare and record in the SDF test procedures for conducting each informal Unit test.

5.4.1.7 The contractor shall perform informal Unit tests according to the test plans for informal Unit testing contained in the STP and according to the Unit test cases and Unit test procedures contained in the SDF.

5.4.1.8 The contractor shall record in the SDF the test results of all informal Unit testing.

5.4.1.9 The contractor shall make necessary revisions to the design documentation and code, and shall update the SDFs of all Units that undergo design or coding changes based on Unit tests.

5.4.1.10 The contractor shall enter into the Developmental Configuration and release for integration each coded Unit that has been successfully tested and reviewed (see 5.8.1.2.6).

5.4.1.11 The contractor shall develop detailed test procedures for conducting each informal CSC integration test.

5.4.1.12 The contractor shall prepare preliminary versions of test procedures for conducting each formal CSCI test and for analyzing formal CSCI test results. Test procedures shall include:

- a. Schedule
- b. Pretest procedures, including equipment preparation and software preparation
- c. Each step of the procedures

- d. Applicable data reduction and data analysis procedures
- e. Assumptions made and constraints imposed on formal test procedures.

5.4.1.13 The contractor shall update with additional known details all information and instructions pertaining to computer system operation, software operation by users, computer system diagnostics, programming or reprogramming software for the target computer, and modifying or replacing firmware (see 5.4.2.7).

5.4.1.14 The contractor shall conduct internal in-process reviews during this phase (see 5.8.1.2.6) and shall make all necessary changes based on the results of the internal reviews.

5.4.2 Products - Coding and Unit Testing. The contractor shall produce the following products during Coding and Unit Testing (see 6.2).

5.4.2.1 The contractor shall produce updated versions of the SDP, SSPM, SCMP, and SQEP, as necessary.

5.4.2.2 The contractor shall produce records and summary reports of the internal reviews conducted (see 5.8.2.1 and 5.8.2.2).

5.4.2.3 The contractor shall produce the source and object code and, as necessary, updated design documentation for each Unit of each CSCI.

5.4.2.4 The contractor shall produce updated SDFs as necessary for all Units (e.g., modified Unit test procedures, retest results, etc.). All SDFs shall be in the standard format described in either the SSPM or SDP (see Appendix D).

5.4.2.5 The contractor shall produce detailed test procedures for conducting each informal CSC integration test. These procedures shall be in the format described in either the SSPM or SDP (see Appendix D).

5.4.2.6 The contractor shall produce a preliminary version of the Software Test Procedure (STPR) to describe the detailed procedures for conducting formal CSCI tests and for analyzing formal CSCI test results.

5.4.2.7 The contractor shall produce updated versions of the CSOM, SUM(s), CSDM, SPM, and FSM.

5.4.3 Developmental Configuration - Coding and Unit Testing. The contractor shall enter any updated design documentation, source and object code, and associated listings for each successfully tested and reviewed Unit into the Developmental Configuration for the CSCI (see 5.8.1.2.6).

5.5 CSC Integration and Testing. The contractor shall integrate Units of code entered in the Developmental Configuration and perform informal tests on aggregates of integrated Units. In order to test critical functions of each CSCI early, formal tests may be conducted during this phase. Formal tests conducted during this phase require: (1) the contractor to complete the applicable formal test procedures, (2) contracting agency approval of the applicable formal test procedures, and (3) the contractor to perform the tests in accordance with the approved test procedures.

5.5.1 Activities - CSC Integration and Testing. The contractor shall perform the following activities during CSC Integration and Testing.

5.5.1.1 The contractor shall monitor the development effort for consistency with the SDP, SSPM, SCMP, and SQEP (see 5.8.1.2.2). The contractor shall notify the contracting agency of proposed changes to these documents and make necessary revisions. All proposed changes shall be subject to disapproval by the contracting agency. In addition, the contractor shall notify the contracting agency at the next review, audit, or in the next status report (whichever comes first) of any actions or procedures occurring during CSC Integration and Testing that deviate from the SDP, SSPM, SCMP, or SQEP.

5.5.1.2 The contractor shall integrate and test aggregates of Units in a top-down sequence, unless alternate methodologies have been proposed in either the SSPM or SDP (see Appendix D) and have received contracting agency approval.

5.5.1.3 The contractor may depart from a top-down approach to: (1) integrate or test critical Units or (2) incorporate commercially available, reusable, and Government furnished software. The contractor shall describe the criteria for determining critical Units in either the SSPM or SDP (see Appendix D). Examples of criteria for determining criticality are software performance, cost, and schedule.

5.5.1.4 As Units are successively integrated with one another, the contractor shall compare memory and processing time values with allocations established during Preliminary and Detailed Design. The contractor shall also compare any system resources affected by the integrated Units with specified requirements (e.g., secondary storage, communication channel utilization, etc.). The contractor shall modify, as necessary, all controlled or baselined documentation based on the memory, processing time, and system resources comparisons. All modifications to controlled or baselined documentation shall be made in accordance with the configuration management requirements contained herein (see 5.7).

5.5.1.5 The contractor shall informally test aggregates of integrated Units according to the test plans contained in the STP and the test cases and test procedures developed in previous

phases.

5.5.1.6 The contractor shall document the results of all integration testing in the standard format described in either the SSPM or SDP (see Appendix D).

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

5.5.1.8 The contractor shall complete preparation of detailed procedures for conducting each formal CSCI test and for analyzing formal test results (see 5.4.1.12).

5.5.1.9 The contractor shall update with additional known details all information and instructions pertaining to computer system operation, software operation by users, computer system diagnostics, programming or reprogramming software for the target computer, and modifying or replacing firmware (see 5.5.2.7).

5.5.1.10 The contractor shall conduct internal in-process reviews during this phase (see 5.8.1.2.7) and shall make all necessary changes based on the results of the internal reviews, prior to presenting the informal test results, completed formal CSGI test procedures, and updated operation and support documentation to the contracting agency.

5.5.2 Products - CSC Integration and Testing. The contractor shall produce the following products during CSC Integration and Testing (see 6.2).

5.5.2.1 The contractor shall produce updated versions of the SDP, SSPM, SCMP, and SQEP, as necessary.

5.5.2.2 The contractor shall produce records and summary reports of the internal reviews conducted (see 5.8.2.1 and 5.8.2.2).

5.5.2.3 The contractor shall produce the source and object code for each complete CSCI by integrating its constituent parts.

5.5.2.4 The contractor shall produce the informal integration test results documented in the standard format described in either the SSPM or SDP (see Appendix D).

5.5.2.5 The contractor shall produce updated design documents and SDFs to reflect changes based on integration testing.

5.5.2.6 The contractor shall produce the completed STPR for each CSCI.

5.5.2.7 The contractor shall produce updated versions of the CSOM, SUM(s), CSDM, SPM, and FSM.

5.5.3 Formal Reviews - CSC Integration and Testing. The contractor shall present informal CSC integration test results and the STPR for each CSCI at a Test Readiness Review (TRR). The contractor shall also present the updated CSOM, SUM(S), and CSDM. The purpose of the TRR is to review the informal test results, formal test procedures, and operation and support documents with the contracting agency, and to demonstrate to the contracting agency that: (1) the STPR is complete, (2) the contractor is ready to begin formal testing, and (3) the updated versions of the CSOM, SUM(s), and CSDM will, in final form, adequately address the operation and support of the computer system. Specific details regarding the TRR process are contained in MIL-STD-1521.

5.5.4 Developmental Configuration - CSC Integration and Testing. The contractor shall enter any updated design documentation, source code, object code, and associated listings into the Developmental Configuration for each CSCI.

5.6 CSCI Testing. The contractor shall conduct formal tests on each CSCI to show that the CSCI satisfies its specified requirements. The contractor shall also record and analyze formal test results. Conducting and analyzing formal tests shall be performed by individuals sufficiently independent from the individuals responsible for development to permit objective testing.

5.6.1 Activities - CSCI Testing. The contractor shall perform the following activities during CSCI Testing.

5.6.1.1 The contractor shall monitor the development effort for consistency with the SDP, SSPM, SCMP, and SQEP (see 5.8.1.2.2). The contractor shall notify the contracting agency of proposed changes to these documents and make necessary revisions. All proposed changes shall be subject to disapproval by the contracting agency. In addition, the contractor shall notify the contracting agency at the next review, audit, or in the next status report (whichever comes first) of any actions or procedures occurring during CSCI Testing that deviate from the SDP, SSPM, SCMP, or SQEP.

5.6.1.2 Individuals sufficiently independent from the individuals responsible for development shall perform formal tests on each CSCI in accordance with the: (1) formal test plans described in the STP, (2) formal test cases described in the STD, and (3) formal test procedures contained in the STPR.

5.6.1.3 Individuals sufficiently independent from the individuals responsible for development shall report the results of all formal CSCI tests. The test reports shall include:

- a. Summary and detail of the test results
- b. Detailed test history
- c. Evaluation of test results, and recommendations

d. Test procedure deviations.

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

5.6.1.5 The contractor shall identify the exact version of each deliverable CSCI and the interim changes occurring between versions. This identification shall include:

- a. Inventory of materials to be released
- b. Inventory of CSCI contents
- c. Class I changes installed
- d. Class II changes installed
- e. Adaptation data
- f. Interface compatibility
- g. Bibliography of reference documents
- h. Operational description
- i. Installation instructions
- j. Possible problems and known errors.

5.6.1.6 The contractor shall complete all information and instructions pertaining to computer system operation, software operation by users, computer system diagnostics, programming or reprogramming software for the target computer, and modifying or replacing firmware (see 5.6.2.7).

5.6.1.7 The contractor shall conduct internal in-process reviews during this phase (see 5.8.1.2.8) and shall make all necessary changes based on the results of the internal reviews, prior to presenting the formal test results and completed operation and support documents to the contracting agency.

5.6.2 Products - CSCI Testing The contractor shall produce the following products during CSCI Testing (see 6.2).

5.6.2.1 The contractor shall produce updated versions of the SDP, SSPM, SCMP, and SQEP, as necessary.

5.6.2.2 The contractor shall produce records and summary reports of the internal reviews conducted (see 5.8.2.1 and 5.8.2.2).

5.6.2.3 The contractor shall produce Software Test Reports (STRs) which document the results of formal CSCI tests, test data analysis, and any deviations or discrepancies discovered in the testing.

5.6.2.4 The contractor shall produce the updated source and object code for each CSCI and prepare them for delivery in accordance with the requirements of the SRS.

5.6.2.5 The contractor shall produce a Software Product

Specification (SPS) for each CSCI, consisting of all the documents and listings comprising the Developmental Configuration for the CSCI. Some CSCIs may require integration with other computer systems, HWCIs, or CSCIs before all formal testing can be completed. In such cases the SPS cannot be completed until after such integration and testing. Additional guidance on preparing specifications is provided in MIL-STD-490.

5.6.2.6 The contractor shall produce a Version Description Document (VDD) for each CSCI.

5.6.2.7 The contractor shall produce completed versions of the CSOM, SUM(s), CSDM, SPM, and FSM.

5.6.3 Audits - CSCI Testing. The contractor shall present the STR(s) for each CSCI and the CSOM, SUM(s), and CSDM at a Functional Configuration Audit (FCA). The contractor shall present the SPS, VDD, and source and object code for each CSCI at a Physical Configuration Audit (PCA). The contractor shall also present the CSOM, SUM(s), CSDM, SPM, and FSM at the PCA. The purpose of the FCA is to demonstrate to the contracting agency that the CSCI was successfully tested and meets the requirements of the SRS and the IRS(s). The FCA also demonstrates to the contracting agency that the CSOM, SUM(s), and CSDM adequately address the operation and support of the computer system. The purpose of the PCA is to demonstrate to the contracting agency that the SPS is complete and reflects an up-to-date technical description of the CSCI. FCA and PCA for the CSCI may be postponed until the system level, if formal testing of the CSCI requires system level integration. Specific details regarding the FCA and PCA processes are contained in MIL-STD-1521.

5.6.4 Baselines - CSCI Testing. The configuration identification documents for the HWCIs and CSCIs that comprise a system form a single Product Baseline. Upon successful completion of the FCA and PCA for each CSCI and when authenticated by the contracting agency, the SPS for the CSCI will be entered into the Product Baseline. Upon SPS entry into the Product Baseline, the CSCI's Developmental Configuration shall cease to exist. Specific details regarding the baseline process are contained in MIL-STD-483 and MIL-STD-490.

5.6.5 Software acceptance. Acceptance of the software by the contracting agency depends on the nature of the end items under contract. If only software is contracted for, then software acceptance follows PCA for each CSCI. If an integrated hardware and software system is contracted for, then software acceptance is part of system acceptance and follows system level PCA. Software acceptance shall be predicated on the following:

- a. Satisfaction of criteria specified in the SOW and contract
- b. Satisfactory completion of FCA and PCA

- c. Number and severity of unresolved software and documentation errors
- d. Documented evidence of correlation between the source code and object code
- e. Consistency between the code and its associated SPS and VDD
- f. Contractor recommendations for acceptance to the contracting agency or its designated representative
- g. Certification of compliance with contractual requirements.

Specific details regarding the software acceptance process are contained in MIL-STD-1521.

5.6.6 Installation and checkout. If required by the SOW, the contractor shall install and checkout the deliverable software at Government-designated facilities. The contractor shall specify the installation and checkout procedures to be followed in the SDP.

5.7 Configuration Management (CM). The contractor shall implement the procedures described in either the SCMP, SDP, or system CM plan (see Appendix D) which provide technical and administrative direction and surveillance to: (1) identify and document the functional and physical characteristics of each CSCI, (2) control changes to those characteristics, and (3) record and report the processing of changes and the status of implementation. The contractor shall perform software configuration management within the framework of the system configuration management and shall ensure that integrated procedures address the total system requirements, including such items as hardware, related CSCIs, support and training elements and facilities, and Government furnished hardware or software, as applicable. The contractor shall perform configuration management on all non-deliverable software used in the development and on revisions to commercially-available computer resources, as described in either the SCMP, SDP, or system CM plan (see Appendix D). The contractor is encouraged to use automated tools in performing configuration management (see 5.9.1.4). Additional guidance on configuration management practices and baselines may be found in MIL-STD-483 and MIL-STD-490.

5.7.1 Activities - Configuration Management. The contractor shall perform the following configuration management activities.

5.7.1.1 Configuration identification. The contractor shall implement the procedures specified in either the SCMP, SDP, or system CM plan (see Appendix D) and approved by the contracting agency. These procedures shall identify the various TLCSCs, LLCSCs, and Units that make up the CSCI, and shall indicate the relationship between the CSCI elements and the documentation for

the CSCI. Configuration identification by the contractor shall include the following activities.

5.7.1.1.1 The contractor shall identify the following documentation which establishes and defines:

- a. The Functional and Allocated Baselines, which shall consist of system and CSCI requirements documents provided or approved by the contracting agency.
- b. The Developmental Configuration, which consists of documentation defining the design and code (including revisions) for each CSCI and its constituent TLCSCs, LLCSCs, and Units. The Developmental Configuration also contains the complete and current software code (source and object) of all Units that have been successfully tested and reviewed. Documentation and code comprising the Developmental Configuration shall be designated for configuration control by the contractor until the documentation is entered into the Product Baseline and the source and object code are delivered. Documentation and code shall be provided to the contracting agency for information or provisional review in accordance with the contract data requirements.
- c. The Product Baseline which will be established upon successful completion of FCA and PCA. The Product Baseline will include the approved Developmental Configuration documentation for each CSCI and shall be under contracting agency configuration control, unless otherwise stipulated in the contract.

5.7.1.1.2 The contractor shall identify all documentation and computer software media containing code, documentation, or both by titling, labeling, numbering, and cataloging procedures. The procedures shall accomplish the following:

- a. Uniquely identify all the TLCSCs, LLCSCs, and Units of each CSCI, and the specific versions of each element to which a document applies.
- b. Uniquely identify the serial, edition, change status, and other identification details of each document.
- c. Identify the specific contents of each medium, including change status.

5.7.1.2 Configuration control. The contractor shall implement the procedures specified in either the SCMP, SDP, or system CM plan (see Appendix D) and approved by the contracting agency, to control all changes to the Developmental Configuration, formally baselined documents, and code for each CSCI. Configuration control by the contractor shall include the following activities.

5.7.1.2.1 The contractor shall include under internal configuration control all items entered into the contractor's Developmental Configuration.

5.7.1.2.2 The contractor shall form a Software Configuration Control Board (SCCB) that shall have control over the Developmental Configuration. No changes shall be made to the Developmental Configuration without SCCB approval.

5.7.1.2.3 The contractor shall implement a corrective action system to report and track all problems and to implement necessary changes (see 5.8.1.10).

5.7.1.2.4 Proposed changes which impact the approved documentation comprising the Functional, Allocated, or Product Baselines shall be classified and processed in accordance with DOD-STD-480 or MIL-STD-481, as contractually specified, and shall be subject to contracting agency approval prior to implementation.

5.7.1.2.5 The contractor shall control the preparation and dissemination of changes to both the software and its documentation to reflect approved and implemented changes.

5.7.1.3. Configuration status accounting. The contractor shall implement the procedures specified in either the SCMP, SDP, or system CM plan (see Appendix D) and approved by the contracting agency, to generate periodic status reports on all products in the Developmental Configuration and in the Allocated and Product Baselines. Status reports shall: (1) provide traceability of changes to controlled products, (2) serve as a basis for communicating the status of configuration identifications and associated software, and (3) serve as a vehicle for ensuring that delivered documents describe and represent the associated software.

5.7.2 Products - Configuration Management. The contractor shall prepare the following products of configuration management (see 6.2).

5.7.2.1. The contractor shall prepare a software problem or change report to describe each problem discovered and the associated proposed change. All such reports shall be in the format specified in either the SCMP, SDP, or system CM plan (see Appendix D).

5.7.2.2 The contractor shall prepare an Engineering Change Proposal (ECP) in accordance with DOD-STD-480 or MIL-STD-481, as contractually specified, to propose each change to the Government that impacts the CSCI's cost, schedule, interfaces, or Government-controlled baselines.

5.7.2.3 The contractor shall prepare a Specification Change Notice (SCN) in accordance with MIL-STD-490 to describe changes to

Government-controlled baselines. Preliminary SCNs shall accompany ECPs, as applicable. Additional guidance may be found in MIL-STD-483 and MIL-STD-490.

5.7.2.4 The contractor shall prepare a Version Description Document (VDD) to identify new and interim versions of each CSCI and associated software product specifications entered in the Product Baseline.

5.7.2.5 The contractor shall provide the contracting agency with CSCI configuration information from the status accounting system, in the form of reports, electronic data transmittal, or other media, as contractually required.

5.7.3 Audits - Configuration Management. The contracting agency will conduct, and the contractor shall support, an FCA and PCA of each CSCI in accordance with MIL-STD-1521.

5.8 Software Quality Evaluation. The contractor shall establish and implement internal procedures to: (1) evaluate the requirements established for the software, (2) evaluate the methodologies established and implemented for developing the software, (3) evaluate the products of the software development process, (4) provide feedback and recommendations based on these evaluations that can be used to effect improvements in the software quality, and (5) perform corrective action in terms of detecting, reporting, and tracking problems with controlled software and documentation. The methods of evaluation (e.g., sampling) shall be specified by the contractor in either the SQEP or the SDP.

5.8.1 Activities - Software Quality Evaluation. The contractor shall perform the following software quality evaluation activities.

5.8.1.1 Planning. The contractor shall perform the planning necessary to establish and implement the tasks specified in Section 5.8 herein.

5.8.1.2 Internal reviews. The contractor shall conduct internal reviews of the methodologies proposed in the contractor's planning documents and of their implementation on the software development project. These reviews shall evaluate the compliance of proposed methodologies with this standard, their adequacy to produce software products that will meet established requirements, and compliance of the software development process with established methodologies. In addition, the contractor shall conduct internal in-process reviews of the software development products. The internal reviews in each software development phase shall be as follows.

5.8.1.2.1 Evaluation criteria. In conducting reviews of software and documentation, the contractor shall use the following

evaluation criteria in addition to those specified in 5.8.1.2.2 through 5.8.1.2.8:

- a. Adherence to required format
- b. Compliance with contractual requirements
- c. Internal consistency
- d. Understandability
- e. Technical adequacy
- f. Degree of completeness appropriate to the phase.

5.8.1.2.2 Internal reviews - all phases. The contractor shall conduct the following internal reviews during all phases of the software development cycle:

- a. Review the newly prepared or revised SDP, SSPM, SCMP, and SQEP for the criteria identified in 5.8.1.2.1, compliance with this standard, and consistency with one another.
- b. Review the activities and the tools, procedures, and methodologies employed during the phase for consistency with the contractor's software development plans. Included in this review shall be evaluation of: (1) software configuration management, (2) software development library, (3) documentation control, (4) storage and handling of project media, (5) control of non-deliverables, (6) risk management, (7) corrective action, and (8) conformance to all approved standards and procedures.

5.8.1.2.3 Internal review - Software Requirements Analysis. The contractor shall conduct internal reviews during Software Requirements Analysis. In addition to the reviews specified in 5.8.1.2.2, the contractor shall:

- a. Review the OCD for: (1) the criteria in 5.8.1.2.1, (2) consistency with the SSS, and (3) ability to provide a high-level understanding of the system.
- b. Review the evolving requirements and the SRS and IRS(s) for: (1) the criteria in 5.8.1.2.1, (2) traceability of the software requirements to the system/segment, prime item, or critical item specification requirements, (3) consistency of the interface requirements with specifications for interfacing elements, (4) consistency of the SRS and IRS(s) with one another, and (5) testability of the software functional, performance, and interface requirements.

5.8.1.2.4 Internal review - Preliminary Design. The contractor shall conduct internal reviews during Preliminary Design. In addition to the reviews specified in 5.8.1.2.2, the contractor shall:

- a. Review the evolving top-level design and STLDD for: (1) the criteria in 5.8.1.2.1, (2) traceability to software

requirements, (3) use of appropriate design techniques, and (4) appropriate level of detail.

- b. Review the STP for: (1) the criteria in 5.8.1.2.1, (2) adequate test coverage of all software requirements, (3) consistency with the software development plans, and (4) adequacy of test planning.
- c. Review the preliminary versions of the CSOM, SUM(s), and CSDM for: (1) the criteria in 5.8.1.2.1, (2) consistency with software requirement specifications and design documents, (3) appropriateness of content for operators or users, and (4) consistency with one another.
- d. Review the preliminary CRISD for: (1) the criteria in 5.8.1.2.1, (2) consistency with the Government's support concepts, and (3) adequacy of support planning.

5.8.1.2.5 Internal review - Detailed Design. The contractor shall conduct internal reviews during Detailed Design. In addition to the reviews specified in 5.8.1.2.2, the contractor shall:

- a. Review the evolving detailed design and the SDDD, IDD(s), and DBDD(s), as applicable, for: (1) the criteria in 5.8.1.2.1, (2) traceability to software requirements specifications and top-level design documentation, (3) use of appropriate design techniques, and (4) consistency with one another.
- b. Review the STD for: (1) the criteria in 5.8.1.2.1, (2) traceability to the STP, (3) adequate test coverage of the software requirements, and (4) consistency with design documentation.
- c. Review a representative subset of the software development files for: (1) the criteria in 5.8.1.2.1, and (2) accuracy of schedule and status information. Review Unit test cases for: (1) the criteria in 5.8.1.2.1, (2) traceability to the STP, (3) adequate test coverage of Unit requirements, and (4) consistency with the design documentation.
- d. Review the CSC integration test cases for: (1) the criteria in 5.8.1.2.1, (2) traceability to the STP, (3) adequate test coverage of the software requirements, and (4) consistency with design documentation.
- e. Review the updated CSOM, SUM(s), and CSDM for: (1) the criteria in 5.8.1.2.1, (2) consistency with software requirement and design documents, (3) appropriateness of content for operators or users, and (4) consistency with one another.

- f. Review the completed CRISD for: (1) the criteria in 5.8.1.2.1, (2) consistency with the Government's support concepts, and (3) adequacy of support planning.
- g. Review the SPM and FSM for: (1) the criteria in 5.8.1.2.1, (2) consistency with design documentation, and (3) appropriateness of content for support personnel.

5.8.1.2.6 Internal review - Coding and Unit Testing. The contractor shall conduct internal reviews during Coding and Unit Testing. In addition to the reviews specified in 5.8.1.2.2, the contractor shall:

- a. Review the evolving and completed source code of each software Unit for: (1) the criteria in 5.8.1.2.1, (2) compliance with coding standards, and (3) traceability to detailed design documentation.
- b. Review a representative subset of the updated software development files for: (1) the criteria in 5.8.1.2.1, and (2) the accuracy of status and schedule information. Review the Unit test procedures and Unit test results for: (1) the criteria in 5.8.1.2.1, and (2) traceability to Unit test plans and Unit test cases. Based on Unit test results, evaluate whether each Unit is ready to be entered into the Developmental Configuration.
- c. Review the updated STLDD, SDDD, IDD(s), and DBDD(s), as applicable, for: (1) the criteria in 5.8.1.2.1, (2) traceability to software requirements specifications, (3) use of appropriate design techniques, and (4) consistency with one another.
- d. Review, as applicable, updated source code for: (1) the criteria in 5.8.1.2.1, (2) compliance with coding standards, and (3) consistency with the updated detailed design documentation.
- e. Review the informal CSC integration test procedures for: (1) the criteria in 5.8.1.2.1, (2) traceability to CSC integration test plans and test cases, (3) adequate test coverage of software requirements, and (4) consistency with design documents.
- f. Review the preliminary STPR for: (1) the criteria in 5.8.1.2.1, (2) traceability to the STP and STD, (3) adequate test coverage of the software requirements, and (4) consistency with the design documentation.
- g. Review the updated CSOM, SUM(s), and CSDM for: (1) the criteria in 5.8.1.2.1, (2) consistency with software requirement and design documents, (3) appropriateness of content for operators or users, and (4) consistency with one

another.

- h. Review, as applicable, the updated SPM and FSM for: (1) the criteria in 5.8.1.2.1, (2) consistency with design documentation, and (3) appropriateness of content for support personnel.

5.8.1.2.7 Internal review - CSC Integration and Testing. The contractor shall conduct internal reviews during CSC Integration and Testing. In addition to the reviews specified in 5.8.1.2.2, the contractor shall:

- a. Review the informal test results of CSC integration testing for: (1) the criteria in 5.8.1.2.1, and (2) traceability to the CSC test cases and test procedures. Based on the informal integration test results, evaluate whether the integrated CSCI performs correctly and is ready to undergo formal testing.
- b. Review the updated STLDD, SDDD, IDD(s), and DBDD(s), as applicable, for: (1) the criteria in 5.8.1.2.1, (2) traceability to software requirements specifications, (3) use of appropriate design techniques, and (4) consistency with one another.
- c. Review updated source code for: (1) the criteria in 5.8.1.2.1, (2) compliance with coding standards, and (3) consistency with the updated design documentation.
- d. Review a representative subset of the updated software development files, as applicable, for: (1) the criteria in 5.8.1.2.1, and (2) accuracy of status and schedule information.
- e. Review the completed STPR for: (1) the criteria in 5.8.1.2.1, (2) traceability to the STP and STD, (3) adequate test coverage of the software requirements, and (4) consistency with the design documentation.
- f. Review the updated CSOM, SUM(s), and CSDM for: (1) the criteria in 5.8.1.2.1, (2) consistency with software requirement and design documents, (3) appropriateness of content for operators or users, and (4) consistency with one another.
- g. Review, as applicable, the updated SPM and FSM for: (1) the criteria in 5.8.1.2.1, (2) consistency with design documentation, and (3) appropriateness of content for support personnel.

5.8.1.2.8 Internal review - CSCI Testing. The contractor shall conduct internal reviews during CSCI Testing. In addition to the reviews specified in 5.8.1.2.2, the contractor shall:

- a. Monitor the CSCI testing to ensure that: (1) it is performed using the current controlled version of the code, (2) it is conducted in accordance with approved test plans, descriptions, and procedures, and (3) it includes all necessary retesting.
- b. Review the STRs for: (1) the criteria in 5.8.1.2.1, and (2) traceability of the CSCI test results to the CSCI test plans, test cases, and test procedures. Based on the CSCI test results, evaluate whether the CSCI meets its specified requirements.
- c. Review the updated STLDD, SDDD, IDD(s), and DBDD(s), as applicable, for: (1) the criteria in 5.8.1.2.1, (2) traceability to software requirements specifications, (3) use of appropriate design techniques, and (4) consistency with one another.
- d. Review updated source code, as applicable, for: (1) the criteria in 5.8.1.2.1, (2) compliance with coding standards, and (3) consistency with the updated detailed design documentation.
- e. Review a representative subset of updated software development files, as applicable, for: (1) the criteria in 5.8.1.2.1, and (2) accuracy of status and schedule information.
- f. Review the SPS for: (1) the criteria in 5.8.1.2.1, and (2) incorporation of design documentation and software listings consistent with the "as-built" software.
- g. Review the VDD for: (1) the criteria in 5.8.1.2.1, and (2) accuracy in reflecting the exact version of each CSCI.
- h. Review the completed CSOM, SUM(s), and CSDM for: (1) the criteria in 5.8.1.2.1, (2) consistency with the SPS, (3) appropriateness of content for operators or users, and (4) consistency with one another.
- i. Review, as applicable, the updated SPM and FSM for: (1) the criteria in 5.8.1.2.1, (2) consistency with design documentation, and (3) appropriateness of content for support personnel.

5.8.1.3 Formal reviews and audits. The contractor shall evaluate the planning and preparation performed for each formal review and audit in 5.1 through 5.6, to ensure that all required products will be available and ready for Government review.

5.8.1.4 Acceptance inspection. The contractor shall support acceptance inspection by ensuring that all required products are available and ready for Government inspection, all required

procedures have been performed, and evidence of these procedures is available for Government inspection.

5.8.1.5 Installation and checkout. The contractor shall evaluate installation and checkout of the software, if required by the contract, to ensure that this activity has been carried out in compliance with procedures specified in the software development plans.

5.8.1.6 Evaluation of subcontractor products. Prior to accepting software or documentation developed by subcontractors, the contractor shall evaluate them for completeness, technical adequacy, and compliance with subcontract requirements.

5.8.1.7 Commercially available, reusable, and Government furnished software. The contractor shall evaluate the planning performed for the use of commercially available, reusable, and Government furnished software to ensure that all relevant factors have been considered. Upon acquisition, the contractor shall evaluate the software to determine whether it performs as documented, prior to incorporating it into the software being developed. The contractor shall certify that commercially available and reusable software performs as documented and that it is documented adequately.

5.8.1.8 Preparation of quality records. The contractor shall prepare and maintain records of each quality evaluation performed. These records shall identify the date of the evaluation, evaluation participants, items or activities reviewed, objectives of the evaluation, all detected problems, and any recommendations resulting from the evaluation.

5.8.1.9 Quality reporting. The contractor shall prepare reports that provide to contractor management the results and recommendations from the quality evaluations specified herein. The quality evaluation reports shall identify the activities performed, all detected problems, necessary remedial action, identified trends in the problems reported, and recommended changes to improve software quality.

5.8.1.10 Corrective action system. The contractor shall implement a corrective action system for all software and documentation that has been placed under contractor or Government control (e.g., development plans, test documentation, design documentation, etc.). The corrective action system shall include provisions for: (1) reporting detected problems, (2) analyzing these problems, (3) classifying problems by category and by priority, (4) identifying necessary corrective action, (5) identifying trends in the problems reported, (6) analyzing these trends to recommend changes that will improve software quality, (7) authorizing the implementation of corrective steps, (8) documenting the corrective actions taken, (9) performing reevaluation after corrections have been made, (10) tracking and

closing out the problems reported, and (11) providing Government visibility into critical problems based on the categorization and priority schemes and problem/change reports.

5.8.1.11 Quality cost data. The contractor shall collect, analyze, and document data relative to the cost of detecting and correcting errors in all software and documentation that have been placed under contractor or Government control. The specific data to be collected and the analyses to be performed shall be proposed by the contractor in either the SQEP or SDP (see Appendix D) and shall be subject to contracting agency approval.

5.8.2 Products - Software Quality Evaluation (see 6.2).

5.8.2.1 Quality records. The contractor shall prepare and maintain records of each quality evaluation performed.

5.8.2.2 Quality reports. The contractor shall prepare and maintain reports that summarize the results and recommendations of the quality evaluations performed. These reports shall be available for Government review.

5.8.2.3 Certification. The contractor shall collect and make available for Government inspection evidence indicating the compliance with the requirements of the contract of each contract line item delivered under the contract.

5.8.3 Independence. Each activity specified in 5.8 herein shall be performed by individuals who have sufficient responsibility, authority, resources, and independence to accomplish objective evaluation of the products and activities being reviewed. The degree of independence varies with such factors as project complexity and criticality. The contractor shall specify the degree of independence in either the SQEP or SDP (see Appendix D).

5.9 Software project planning and control. The contractor shall implement procedures for planning and controlling the software development project.

5.9.1 Activities - Software project planning and control. The contractor shall perform the following planning and controlling activities.

5.9.1.1 Sizing and timing assessments. The contractor shall derive sizing and timing parameters appropriate for the CSCI, including minimum reserve capacities, and shall develop initial estimates during Software Requirements Analysis of these parameters' values and allowed margins. During the remainder of the development, the contractor shall monitor these parameters and reallocate as necessary to meet requirements specified in the SRS. As Units of code are completed, tested, and successively integrated with one another, the contractor shall measure these sizing and timing parameters, compare these measurements with

estimates, and update overall CSCI sizing and timing records to reflect the results of these measurements. All modifications to controlled or baselined documentation shall be made in accordance with the configuration management requirements contained herein (see 5.7).

5.9.1.2 Status and cost reporting. The contractor shall maintain cost and schedule forecasts, analyses, and reports to at least the CSCI level. These reports shall indicate to the contracting agency predicted and planned progress versus actual progress. Cost reports shall include budgeted versus actual expenditures and shall conform to the Work Breakdown Structure (WBS) applicable to the development effort. Additional guidance for cost and status reporting may be found in MIL-STD-881.

5.9.1.3 Test documentation control. Once the contracting agency approves the STP, STD, and STPR the contractor shall establish internal control over these documents. The contractor shall notify the contracting agency at the next review, audit, or in the next status report (whichever comes first) of any proposed changes to these documents, and shall obtain contracting agency approval before making any of the proposed changes.

5.9.1.4 Software development library (SDL). The contractor shall establish and implement a software development library for controlling all software and associated documentation. Procedures and methodologies for establishing and implementing the SDL shall be specified in the SDP.

5.9.1.5 Risk management. The contractor shall establish and implement the risk management procedures specified in the SDP for controlling risk. The procedures shall include:

- a. Identifying the risk areas of the project and the constituent risk factors in each area.
- b. Assessing the risk factors identified, including the probability of occurrence and the potential damage.
- c. Assigning appropriate resources to reduce the risk factors.
- d. Identifying and analyzing the alternatives available for reducing the risk factors.
- e. Selecting the most promising alternative for each risk factor.
- f. Planning implementation of the selected alternative for each risk factor.
- g. Obtaining feedback to determine the success of the risk reducing action for each risk factor.

6. NOTES

6.1 Intended use. This standard is intended for use during the development and acquisition of MCCS software, as defined in DOD Directive 5000.29. This standard may also be used for non-MCCS software development and acquisition.

6.2 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. Deliverable data required by this standard is cited in the following subparagraphs.

<u>Paragraph No.</u>	<u>Data Requirements Title</u>	<u>Applicable DID No.</u>
5.1, 5.1.1.5, 5.8.1.2.3, 20.4.1, 20.4.2, 20.4.5.2, 30.3.1, 30.3.1.1, 30.3.1.3, 40.6.2.2	System/Segment Specification	DI-CMAN-80008
4.3, 4.4, 4.6, 4.7, 4.8, 5.1, 5.1.1.1, 5.1.1.2, 5.1.1.7, 5.1.2.1, 5.2, 5.2.1.1, 5.2.1.3, 5.2.1.4, 5.2.2.1, 5.3.1.1, 5.3.1.3, 5.3.1.4, 5.3.1.5, 5.3.1.8, 5.3.2.1, 5.3.2.7, 5.4.1.1, 5.4.1.2, 5.4.1.3, 5.4.1.4, 5.4.2.1, 5.4.2.4, 5.4.2.5, 5.5.1.1, 5.5.1.2, 5.5.1.3, 5.5.1.6, 5.5.2.1, 5.5.2.4, 5.6.1.1, 5.6.2.1, 5.6.6, 5.7, 5.7.1.1, 5.7.1.2, 5.7.1.3, 5.7.2.1, 5.8.1.2.2, 5.8.1.11,	Software Development Plan	DI-MCCR-80030

<u>Paragraph No.</u>	<u>Data Requirements Title</u>	<u>Applicable DID No.</u>
5.8.3, 5.9.1.4, 5.9.1.5, 20.4.3, 20.4.5.2, 30.1, 30.2, 30.3.1.1, 30.3.1.2, 40.5.1, 40.6.2.1		
4.3, 5.1, 5.1.1.1, 5.1.1.2, 5.1.2.1, 5.2.1.1, 5.2.2.1, 5.3.1.1, 5.3.2.1, 5.4.1.1, 5.4.2.1, 5.5.1.1, 5.5.2.1, 5.6.1.1, 5.6.2.1, 5.7, 5.7.1.1, 5.7.1.2, 5.7.1.3, 5.7.2.1, 5.8.1.2.2, 40.5.1, 40.6.2.1	Software Configuration Management Plan	DI-MCCR-80009
4.3, 5.1, 5.1.1.1, 5.1.1.2, 5.1.2.1, 5.2.1.1, 5.2.1.4, 5.2.2.1, 5.3.1.1, 5.3.2.1, 5.4.1.1, 5.4.2.1, 5.5.1.1, 5.5.2.1, 5.6.1.1, 5.6.2.1, 5.8.1.2.2, 5.8.1.11, 5.8.3, 40.5.1, 40.6.2.1	Software Quality Evaluation Plan	DI-MCCR-80010
5.1.1.5, 5.1.1.6, 5.1.1.7, 5.1.2.4, 5.1.3, 5.1.4, 5.2, 5.2.1.2, 5.3.3, 5.6.3, 5.8.1.2.3, 5.9.1.1, 20.4.2, 20.4.3, 20.4.5.2, 40.6.2.2	Software Requirements Specification	DI-MCCR-80025
5.1.1.5, 5.1.1.6, 5.1.2.4, 5.1.3, 5.1.4, 5.2, 5.2.1.2, 5.3.2.4, 5.3.3, 5.6.3, 5.8.1.2.3, 20.4.2, 20.4.3, 20.4.5.2, 40.6.2.2	Interface Requirements Specification	DI-MCCR-80026

<u>Paragraph No.</u>	<u>Data Requirements Title</u>	<u>Applicable DID No.</u>
4.3, 4.8, 5.1, 5.1.1.1, 5.1.1.2, 5.1.1.7, 5.1.2.1, 5.2, 5.2.1.1, 5.2.1.3, 5.2.1.4, 5.2.2.1, 5.3.1.1, 5.3.1.3, 5.3.1.4, 5.3.1.5, 5.3.1.8, 5.3.2.1, 5.3.2.7, 5.4.1.1, 5.4.1.2, 5.4.1.3, 5.4.1.4, 5.4.2.1, 5.4.2.4, 5.4.2.5, 5.5.1.1, 5.5.1.2, 5.5.1.3, 5.5.1.6, 5.5.2.1, 5.5.2.4, 5.6.1.1, 5.6.2.1, 5.8.1.2.2, 20.4.3, 30.1, 30.2, 30.3.1.1, 40.5.1, 40.6.2.1.	Software Standards and Procedures Manual	DI-MCCR-80011
5.2.1.2, 5.2.2.3, 5.2.3, 5.2.4, 5.3.3, 5.8.1.2.4, 5.8.1.2.6, 5.8.1.2.7, 5.8.1.2.8, 20.4.3, 40.6.2.2	Software Top Level Design Document	DI-MCCR-80012
5.2.1.3, 5.2.2.4, 5.2.3, 5.3.1.2, 5.3.2.3, 5.3.3, 5.3.4, 5.8.1.2.5, 5.8.1.2.6, 5.8.1.2.7, 5.8.1.2.8, 20.4.3, 40.6.2.2	Software Detailed Design Document	DI-MCCR-80031
5.2.1.3, 5.2.2.4, 5.2.3, 5.3.1.2, 5.3.2.4, 5.3.3, 5.3.4, 5.8.1.2.5, 5.8.1.2.6, 5.8.1.2.7, 5.8.1.2.8, 20.4.3, 40.6.2.2	Interface Design Document	DI-MCCR-80027

DOD-STD-2167

<u>Paragraph No.</u>	<u>Data Requirements Title</u>	<u>Applicable DID No.</u>
5.2.1.3, 5.2.2.4, 5.2.3, 5.3.1.2, 5.3.2.5, 5.3.3, 5.3.4, 5.8.1.2.5, 5.8.1.2.6, 5.8.1.2.7, 5.8.1.2.8, 20.4.3, 40.6.2.2	Data Base Design Document	DI-MCCR-80028
5.6.2.5, 5.6.3, 5.6.4, 5.6.5, 5.8.1.2.8, 20.4.3, 40.6.2.2	Software Product Specification	DI-MCCR-80029
5.6.1.5, 5.6.2.6, 5.6.3, 5.6.5, 5.7.2.4, 5.8.1.2.8, 40.6.2.2	Version Description Document	DI-MCCR-80013
5.2.1.6, 5.2.2.5, 5.2.3, 5.3.1.14, 5.3.2.8, 5.3.3, 5.4.1.7, 5.5.1.5, 5.6.1.2, 5.8.1.2.4, 5.8.1.2.5, 5.8.1.2.6, 5.8.1.2.7, 5.9.1.3, 20.4.3, 40.6.2.3	Software Test Plan	DI-MCCR-80014
5.3.1.14, 5.3.2.8, 5.3.3, 5.6.1.2, 5.8.1.2.5, 5.8.1.2.6, 5.8.1.2.7, 5.9.1.3, 20.4.3, 40.6.2.3	Software Test Description	DI-MCCR-80015
5.4.1.12, 5.4.2.6, 5.5.1.8, 5.5.2.6, 5.5.3, 5.6.1.2, 5.8.1.2.6, 5.8.1.2.7, 5.9.1.3, 40.6.2.3	Software Test Procedure	DI-MCCR-80016
5.6.1.3, 5.6.2.3, 5.6.3, 5.8.1.2.8, 40.6.2.3	Software Test Report	DI-MCCR-80017
5.2.1.7, 5.2.2.6, 5.2.3, 5.3.1.15, 5.3.2.9, 5.3.3,	Computer System Operator's Manual	DI-MCCR-80018

<u>Paragraph No.</u>	<u>Data Requirements Title</u>	<u>Applicable DID No.</u>
5.4.1.13, 5.4.2.7, 5.5.1.9, 5.5.2.7, 5.5.3, 5.6.1.6, 5.6.2.7, 5.6.3, 5.8.1.2.4, 5.8.1.2.5, 5.8.1.2.6, 5.8.1.2.7, 5.8.1.2.8, 20.4.3, 40.6.2.4	Software User's Manual	DI-MCCR-80019
5.2.1.8, 5.2.2.6, 5.2.3, 5.3.1.15, 5.3.2.9, 5.3.3, 5.4.1.13, 5.4.2.7, 5.5.1.9, 5.5.2.7, 5.5.3, 5.6.1.6, 5.6.2.7, 5.6.3, 5.8.1.2.4, 5.8.1.2.5, 5.8.1.2.6, 5.8.1.2.7, 5.8.1.2.8, 20.4.3, 40.6.2.4	Computer System Diagnostic Manual	DI-MCCR-80020
5.2.1.9, 5.2.2.6, 5.2.3, 5.3.1.15, 5.3.2.9, 5.3.3, 5.4.1.13, 5.4.2.7, 5.5.1.9, 5.5.2.7, 5.5.3, 5.6.1.6, 5.6.2.7, 5.6.3, 5.8.1.2.4, 5.8.1.2.5, 5.8.1.2.6, 5.8.1.2.7, 5.8.1.2.8, 20.4.3, 40.6.2.4	Software Programmer's Manual	DI-MCCR-80021

<u>Paragraph No.</u>	<u>Data Requirements Title</u>	<u>Applicable DID No.</u>
5.3.1.18, 5.3.2.11, 5.3.3, 5.4.1.13, 5.4.2.7, 5.5.1.9, 5.5.2.7, 5.6.1.6, 5.6.2.7, 5.6.3, 5.8.1.2.5, 20.4.3, 40.6.2.4	Firmware Support Manual	DI-MCCR-80022
5.1, 5.1.1.3, 5.1.1.4, 5.1.2.2, 5.1.3, 5.8.1.2.3, 20.4.2, 20.4.3, 20.4.5.2, 40.6.2.4	Operational Concept Document	DI-MCCR-80023
5.2.1.10, 5.2.2.6, 5.2.3, 5.3.1.16, 5.3.2.10, 5.3.3, 5.8.1.2.4, 5.8.1.2.5, 20.4.3, 40.6.2.4	Computer Resources Integrated Support Document	DI-MCCR-80024
5.7, 5.7.1.2, 5.7.1.3, 5.7.2.1	Configuration Management Plan	DI-E-3108
5.7.2.2, 5.7.2.3, 40.6.2.2	Engineering Change Proposal	DI-E-3128
5.7.2.3, 40.6.2.2	Specification Change Notice	DI-E-3134

(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.)

6.3 Subject term (key word) listing.

Acquisition
Code
Code and unit testing
Computer
Computer resources
Computer software
Computer software component
Computer software configuration item
Configuration item
Configuration management
CSC

CSC integration and testing
CSCI
CSCI testing
Data item descriptions
Detailed design
Firmware
Formal testing
Informal testing
LLCSC
Lower level computer software component
Mission-critical
Mission-critical computer resources
Mission-critical computer system
Preliminary design
Quality
Quality evaluation
Requirements analysis
Risk management
Software
Software acquisition
Software code
Software configuration item
Software configuration management
Software design
Software detailed design
Software development
Software integration
Software preliminary design
Software quality
Software quality evaluation
Software requirements
Software requirements analysis
Software standards
Software test
Tailoring
Tailoring of software requirements
Testing
TLCSC
Top-level computer software component
Unit

LIST OF ACRONYMS AND ABBREVIATIONS

10. General.

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
CM	Configuration Management
CRISD	Computer Resources Integrated Support Document
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
CSDM	Computer System Diagnostic Manual
CSOM	Computer System Operator's Manual
DBDD	Data Base Design Document
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
GFE	Government Furnished Equipment
GFS	Government Furnished Software
HOL	Higher order language
HWCI	Hardware Configuration Item
IDD	Interface Design Document
IRS	Interface Requirements Specification
LLCSC	Lower-level computer software component
MCCS	Mission-Critical Computer System
NSCCA	Nuclear safety cross-check analysis
OCD	Operational Concept Document
PCA	Physical Configuration Audit
PDR	Preliminary Design Review
PROM	Programmable read-only memory
RFP	Request for proposal
ROM	Read-only memory
SCCB	Software Configuration Control Board
SCMP	Software Configuration Management Plan
SCN	Specification Change Notice
SDDD	Software Detailed Design Document
SDF	Software Development File
SDL	Software Development Library
SDP	Software Development Plan
SOW	Statement of Work
SPM	Software Programmer's Manual
SPS	Software Product Specification

DOD-STD-2167
APPENDIX A

SQEP	Software Quality Evaluation Plan
SRS	Software Requirements Specification
SSA	Software Support Agency
SSPM	Software Standards and Procedures Manual
SSR	Software Specification Review
SSS	System/Segment Specification
STD	Software Test Description
STLDD	Software Top Level Design Document
STP	Software Test Plan
STPR	Software Test Procedure
STR	Software Test Report
SUM	Software User's Manual
TLCSC	Top-level computer software component
TRR	Test Readiness Review
VDD	Version Description Document
WBS	Work Breakdown Structure

SYSTEM LIFE CYCLE

20. General.

20.1 Purpose. This appendix provides information on the system life cycle and the framework in which software development is conducted under the provisions of this standard.

20.2 Scope. This appendix briefly describes a typical system life cycle and its relationship to iterations of the software development cycle (see Figures 1 and 4). It also describes the documents that result from early system acquisition activities. The activities and phases described in this appendix include activities and phases for which the contractor is not responsible, as well as those for which the contractor is responsible.

20.3 Applicability. The information in this appendix is of a general, tutorial nature and is not a requirement of this standard.

20.4 General information. The system life cycle consists of four phases: Concept Exploration, Demonstration and Validation, Full Scale Development, and Production and Deployment. The software development cycle consists of six phases: Software Requirements Analysis, Preliminary Design, Detailed Design, Coding and Unit Testing, CSC Integration and Testing, and CSCI Testing. The total software development cycle or a subset may be performed within each of the system life cycle phases. Successive iterations of software development usually build upon the products of previous iterations (see Figure 2).

20.4.1 Concept Exploration. The Concept Exploration Phase is the initial planning period when the technical, strategic, and economic bases are established through comprehensive studies, experimental development, and concept evaluation. This initial planning may be directed toward refining proposed solutions or developing alternative concepts to satisfy a required operational capability.

- a. During this phase, proposed solutions are refined or alternative concepts are developed using feasibility assessments, estimates (cost and schedule, intelligence, logistics, etc.), trade-off studies, and analyses. The SSA and user should be involved in these activities.
- b. For computer resources, the software development cycle should be tailored for use during this phase and may result in demonstration of critical algorithms, breadboards, etc.

DOD-STD-2167
APPENDIX B

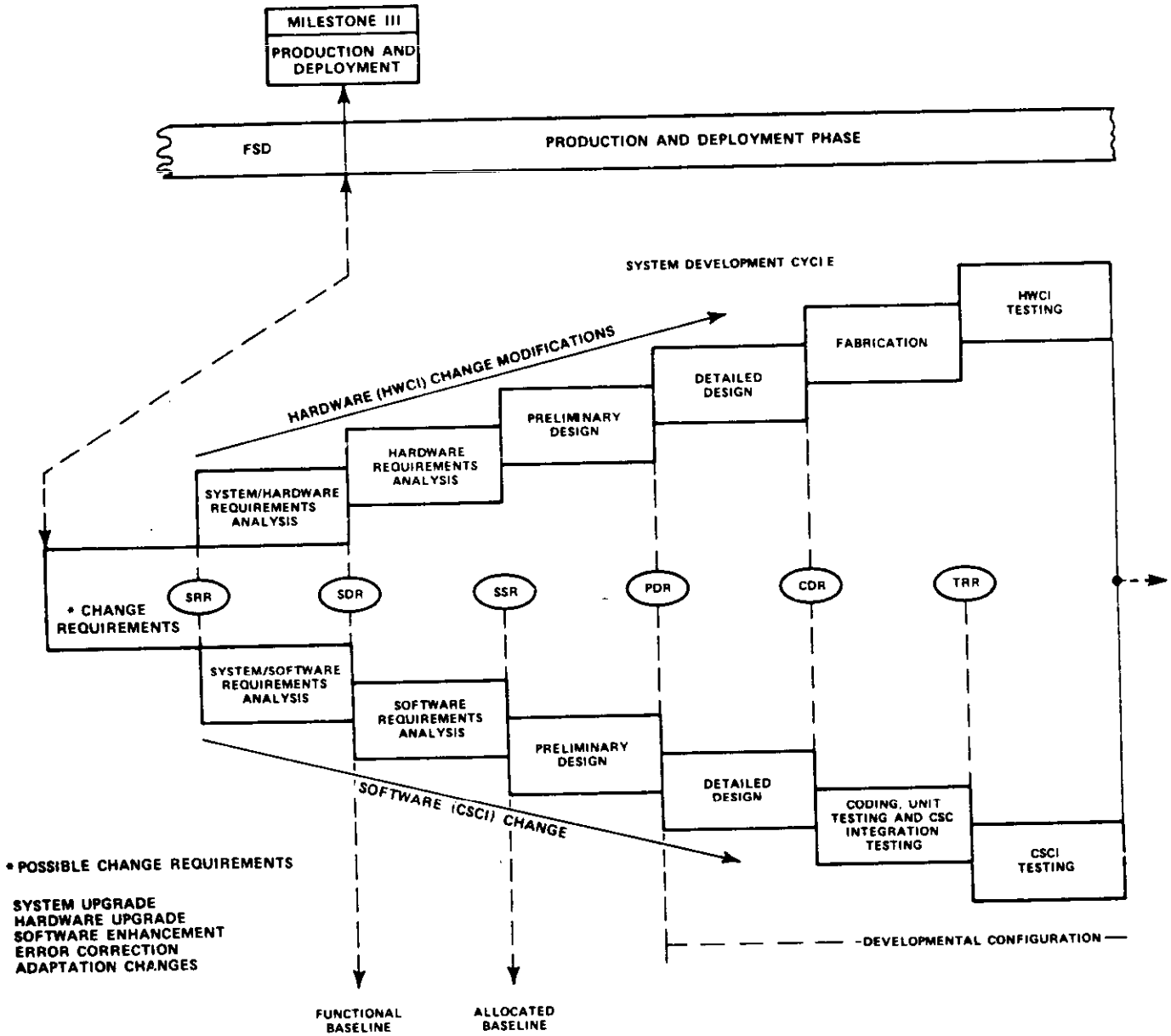


FIGURE 4. System support cycle within the system life cycle.

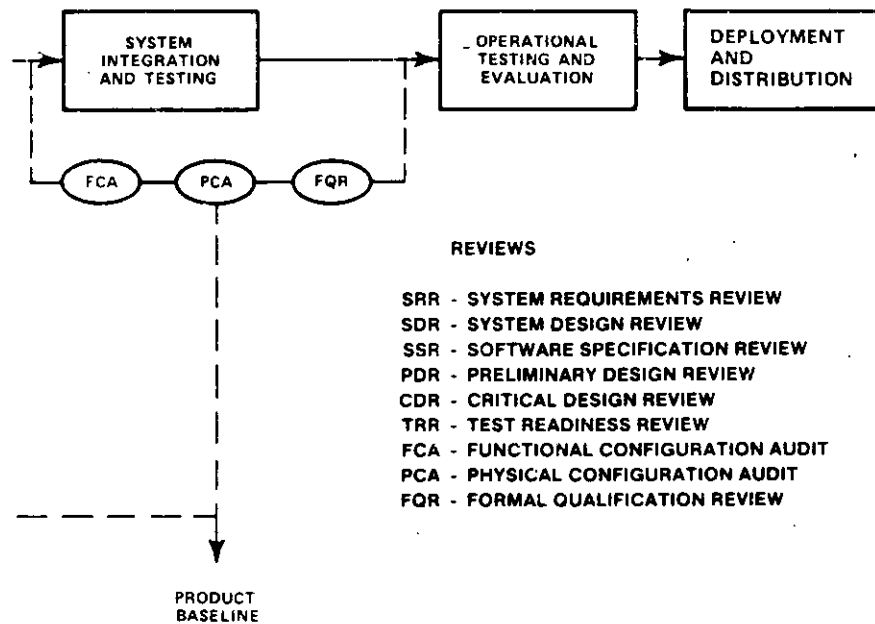
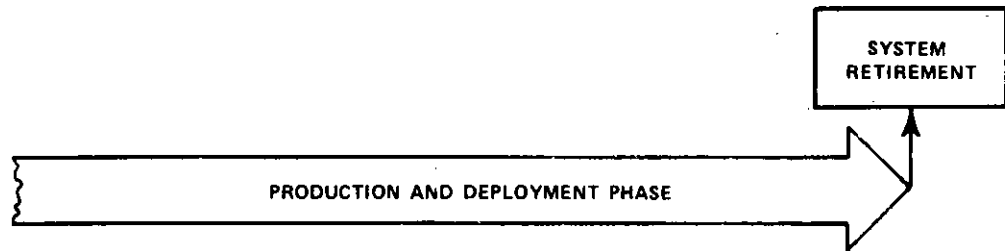


FIGURE 4. System support cycle within the system life cycle. (continued)

- c. The major document resulting from this phase is the initial SSS, which documents total system requirements. The SSS may differentiate between the requirements to be met by computer software and those applicable to hardware design. When applicable, definitions of interfaces between computer equipment functions, communication functions, and personnel functions are provided to enable the further definition and management of the computer software and computer equipment resources. Normally, this information is derived from system engineering studies. Deliverable products at the end of the Concept Exploration phase typically include preliminary SSS(s), preliminary Prime Item Development Specifications, software listings, and software test results, etc. The System Requirements Review is the technical review that should be accomplished.

20.4.2 Demonstration and Validation. The Demonstration and Validation Phase is the period when major system characteristics are refined through studies, system engineering, development of preliminary equipment and prototype computer software, and test and evaluation. The objectives are to validate the choice of alternatives and to provide the basis for determining whether or not to proceed into the next phase.

- a. During this phase, system requirements, including requirements for computer resources, are further defined, and preferred development methodologies for computer software and data bases are selected. The results of validation activities are used to define the system characteristics (performance, cost, and schedule) and to provide confidence that risks have been resolved or minimized.
- b. For computer resources, the software development cycle should be tailored for use during this phase, resulting in prototype software items.
- c. The major documents resulting from this phase are the authenticated SSS(s), authenticated Prime Item Development Specifications, and preliminary IRS(s) and SRSs for each CSCI. The authenticated SSS(s) establish the system or segment Functional Baseline. Each authenticated Prime Item Development Specification contains the system requirements allocated to the equipment and software and establishes the Allocated Baseline for each prime item. Each preliminary SRS contains system or prime item requirements allocated to a CSCI. Each preliminary IRS defines the interfaces and qualification requirements for a CSCI within the system, segment, or prime item. The Allocated Baseline for each CSCI is established following Software Requirements Analysis within the software development cycle. A preliminary version of the Operational Concept Document (OCD) should

also be prepared to identify and describe the mission of the system, operational and support environments of the system, and the functions and characteristics of the computer system within the overall system. The System Design Review is the technical review that should be accomplished.

20.4.3 Full Scale Development. The Full Scale Development phase is the period when the system, equipment, computer software, facilities, personnel subsystems, training, and the principal equipment and software items necessary for support are designed, fabricated, tested, and evaluated. It includes one or more major iterations of the software development cycle. The intended outputs are a system which closely approximates the production item, the documentation necessary to enter the system's Production and Deployment phase, and the test results that demonstrate that the system to be produced will meet the stated requirements. During this phase the requirements for additional software items embedded in or associated with the equipment items may be identified. These requirements may encompass firmware, test equipment, environment simulation, mission support, development support, and many other kinds of software.

- a. Software requirements analysis is performed in conjunction with system engineering activities related to equipment preliminary design. SRSs and IRSSs for each CSCI are completed and authenticated at the SSR, establishing the Allocated Baseline for each CSCI. Requirements for software that is part of an HWCI may be authenticated during HWCI design reviews. The OCD is completed and reviewed at the SSR as well.
- b. A preliminary design effort is accomplished and results in a design approach. For computer software, preliminary design includes the definition of TLCSCs in terms of functions, external and internal interfaces, storage and timing allocation, operating sequences, and data base design. Detailed design of critical lower-level elements of the CSCI may be performed as well. A PDR is held to review the software top-level design document against the respective authenticated specifications for each equipment item and CSCI. The following documents are also presented at the PDR:

STP - to define the plans for informal and formal testing of the CSCI.

Preliminary CSOM - to define the procedures and information necessary to operate the computer system in which the CSCIs execute.

Preliminary SUM(s) - to define the instructions for users to execute each CSCI.

DOD-STD-2167
APPENDIX B

Preliminary CSDM - to define the information and procedures necessary to identify a malfunction and instructions to run the diagnostics.

Preliminary CRISD - to define the information that is required to perform life cycle support of the contractually deliverable software.

- c. Formal engineering change control procedures are implemented to prepare, propose, review, approve, implement, and record engineering changes to each Allocated Baseline.
- d. Informal engineering change control by the contractor starts with the establishment of each CSCI's Developmental Configuration. The Developmental Configuration is established at PDR by the STLDD as the repository for the approved design documents, software, and software listings. Following successful completion of FCA and PCA, the documents and listings of the Developmental Configuration are included in the SPS which establishes the Product Baseline. This baseline is used to control the software as it is integrated with other CSCIs and HWCIs.
- e. Following an acceptable PDR for an item, detailed design of that item begins. During this activity, engineering documentation such as drawings, product specifications, test procedures, and descriptions are produced. For computer software, detailed design is accompanied by detailed design documentation of logical flows, functional sequences and relations, formats, constraints, data bases, and incorporation of reused design. The CDR should assure that the recommended design satisfies the requirements of the SRS and, if applicable, IRS(s). At the CDR, the detailed design documents (i.e. SDDD and, if applicable, DBDD(s) and IDD(s)) are reviewed. Equipment/personnel/computer software interfaces should be finalized at this time. A primary product of the CDR for software is the Government and contractor concurrence on the detailed design documents that will be released for coding and Unit testing. Additional documents prepared during detailed design and reviewed at CDR include:

STD - to describe the test cases for all formal testing of the CSCI.

CRISD - to define the information that is required to perform life cycle support of the contractually delivered software.

SPM - to define the information which facilitates programming or reprogramming software for the target computer.

DOD-STD-2167
APPENDIX B

FSM - to define the information necessary to modify or replace the firmware devices in the Mission-Critical Computer system.

- f. Following CDR, software coding and testing, software integration and testing, software formal testing, system integration and testing, and initial operational test and evaluation are conducted. Software coding is performed in accordance with standards and procedures contained in the approved SDP (or SSPM, if applicable). Software testing is performed according to test plans submitted for review at PDR, test descriptions submitted for review at CDR, and test procedures submitted for review at TRR. These activities normally proceed in such a way that testing of selected functions begins early during development and proceeds by adding successive increments to the point where a complete CSCI is subjected to formal testing. Additional test equipment may be required to properly simulate an operational environment to test a CSCI. The scope and realism of software testing may be progressively expanded as additional increments are made available for this purpose. Adequacy of the performance of the software is checked to the maximum extent possible, sometimes through use of simulation, prior to software installation in a field site or operational computer. Nuclear safety cross-check analysis (NSCCA) is also performed on specified computer resource items during this phase. Satisfactory performance of the software for a large operational system may not be completely demonstrated and assessed until completion of system integration and operational test and evaluation of the equipment or of the system. Software that is relatively insensitive to the system's operational environments may be completely demonstrated earlier.
- g. Functional and Physical Configuration Audits are performed on all items of hardware and software. FCA is conducted on the software at the completion of software formal testing. Based on the nature of the software, PCA may be conducted at the completion of software formal testing or after system integration and testing.
- h. Functional and Physical Configuration Audits may be performed at the system level to authenticate the hardware product specification(s) and the software product specification(s) to establish the system Product Baseline. This baseline acts as an instrument for use in diagnosing troubles, adapting the computer resources to environmental and operational requirements of specific site locations, and proposing changes or enhancements.
- i. Planning for transition of the computer resources to the user and the Software Support Agency (SSA) begins early in

DOD-STD-2167
APPENDIX B

this phase. Necessary agreements should be prepared, coordinated, and approved prior to the end of this phase. The SSA and the user should be involved in this planning.

- j. Provisions are made for follow-on support of the equipment and software configuration items and associated documentation. Failure to properly consider these provisions may result in support complications, obsolete documentation, and costly "modernization" programs. This is particularly true where the system is being developed in a phased manner, providing reduced capabilities for early system integration, operation, and evaluation.

20.4.4 Production and Deployment. The Production and Deployment Phase is the combination of two overlapping periods. The production period is from production approval until the last system item is delivered and accepted. The objective is to efficiently produce and deliver effective and supported systems to the user(s). The deployment period commences with delivery of the first operational system item and terminates when the last system items are removed from the operational inventory.

- a. At system transition, the role of the contracting agency normally terminates except for identified residual tasks and phase-out responsibilities. The supporting and using agencies start providing the resources necessary to support the software throughout the Deployment phase.
- b. Follow-on test and evaluation is performed on operational system items as they are deployed, to assess their operational effectiveness and suitability in a deployed configuration and environment.
- c. After a system is in operational use, there are a variety of changes that may take place on the hardware items, software items, or both hardware and software items. Changes to software items may be necessary to remove latent errors, enhance operations, further system evolution, adapt to changes in mission requirements, or incorporate knowledge gained from operational use. Based upon complexity and other factors such as system interfaces, constraints, and priorities, control of the changes may vary from on-site management to complex checks and balances with mandatory security keys and access codes. The authority to change the software must be carefully and specifically delineated, particularly when security, safety, or special nuclear restrictions are involved. The same six phases of the software development cycle are utilized for each change during the Production and Deployment phase (see Figure 4).

20.4.5 Software Development Cycle Application and Documentation. The software development cycle may span more than one system life

cycle phase, or may occur in any one phase. For example, mission simulation software may undergo one iteration of the software development cycle during the Concept Exploration, while mission application software may undergo many iterations of the software development cycle during the Demonstration and Validation, Full Scale Development, and Production and Deployment phases (see Figure 1).

20.4.5.1 The phases in the software development cycle may involve iterations back to previous phases. For example, design may reveal problems which lead to the revision of requirements and reinstitution of certain analyses; checkout may reveal errors in design, which in turn may lead to redesign or requirements revision; etc.

20.4.5.2 Prior to initiating software development during the Full Scale Development and the Production and Deployment phases, documented plans for software development (e.g. SDP); authenticated system, segment, or prime item specifications; and the OCD typically exist. In earlier life cycle phases, such plans may not yet exist. The software development plans include descriptions of all organizations and procedures to be used in the development effort. The system, segment, or prime item specification identifies the requirements of the system, segment, or prime item. In addition, these specifications identify the HWCIs and CSCIs making up the system, segment, or prime item. The OCD identifies and describes the mission of the system, the system operational and support environments, and the functions and characteristics of the computer system within the overall system. The six phases of the software development cycle are discussed below:

- a. Software Requirements Analysis. The purpose of Software Requirements Analysis is to completely define and analyze the requirements for the software. These requirements include the functions the software is required to accomplish as part of the system, segment, or prime item. Additionally, the functional interfaces and the necessary design constraints are defined. During Full Scale Development, and Production and Deployment, this phase typically begins with the release of the SSS, Prime Item Specification(s), Critical Item Specification(s), or Preliminary SRS(s) and IRS(s), and terminates with the successful accomplishment of the SSR. During this phase, analyses and trade-off studies are performed, and requirements are made definitive. The results of this phase are documented and approved requirements for the software. At the initiation of Software Requirements Analysis, plans for developing the software are prepared or reviewed (as applicable).

DOD-STD-2167
APPENDIX B

- b. Preliminary Design. The purpose of Preliminary Design is to develop a design approach which includes mathematical models, functional flows, and data flows. During this phase various design approaches are considered, analysis and trade-off studies are performed, and design approaches selected. Preliminary Design allocates software requirements to TLCSCs, describes the processing that takes place within each TLCSC, and establishes the interface relationship between TLCSCs. Design of critical lower-level elements of each CSCI may also be performed. The result of this phase is a documented and approved top-level design of the software. The top-level design is reviewed against the requirements prior to initiating the detailed design phase.
- c. Detailed Design. The purpose of Detailed Design is to refine the design approach so that each TLCSC is decomposed into a complete structure of LLCSCs and Units. The detailed design approach is provided in detailed design documents and reviewed against the requirements and top-level design prior to initiating the coding phase.
- d. Coding and Unit Testing. The purpose of Coding and Unit Testing is to code and test each Unit of code described in the detailed design documentation. Each Unit of code is reviewed for compliance with the corresponding detailed design description and applicable coding standards prior to establishing internal control of the Unit and releasing it for integration.
- e. CSC Integration and Testing. The purpose of CSC Integration and Testing is to integrate and test aggregates of coded Units. Integration tests should be performed based on documented integration test plans, test descriptions, and test procedures. CSC Integration test results, and CSCI test plans, descriptions, and procedures for testing the fully implemented software are reviewed prior to the next phase of testing.
- f. CSCI Testing. The purpose of CSCI testing is to test the fully implemented CSCI. Testing during this phase concentrates on showing that the software satisfies its specified requirements. Test results should be reviewed to determine whether the software satisfies its specified requirements.

DESIGN AND CODING STANDARDS

30. General.

30.1 Purpose. This appendix specifies default design and coding standards for the contractor. If the contractor has not proposed internal design and coding standards in either the SSPM or SDP (see Appendix D) and received approval, then the design and coding standards in this appendix shall be applied to all code written by the contractor.

30.2 Applicability. This appendix contains design and coding standards generally applicable to all programming languages. However, it does not provide complete design and coding standards for some higher order languages with advanced capabilities (e.g. Ada, PROLOG, etc.). In such cases, the contractor should propose additions to this appendix in either the SSPM or SDP (see Appendix D) and obtain contracting agency approval.

30.3 Detailed requirements.

30.3.1 Higher order language (HOL). All code shall be written in the HOL specified in the SSS.

30.3.1.1 If one or more compilers are specified in the SSS, then all code shall be compiled by the specified compiler(s). Otherwise, all code shall be compiled by the compilers described in either the SDP or the SSPM (see Appendix D).

30.3.1.2 If the higher order language does not contain the control constructs of Section 30.3.2, the contractor shall use the precompiler specified in the SDP. If a precompiler which is acceptable to the contracting agency does not exist, then these control constructs shall be simulated (i.e. code in the language used shall follow the logic shown in figures 5 through 9 without explicitly using the names of the constructs in the code). If language simulation is used, the same form of the simulated constructs shall be uniformly applied throughout the code.

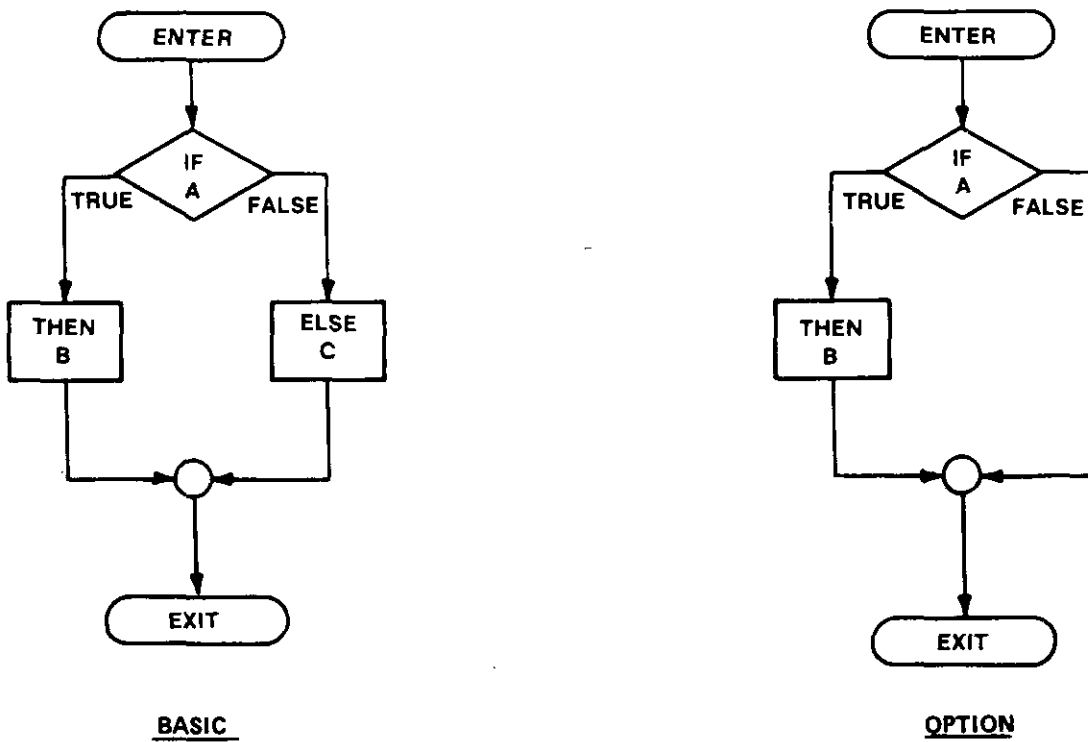
30.3.1.3 A waiver from the contracting agency shall be required in order for the contractor to write code in assembly language or in some HOL other than the HOL specified in the SSS.

30.3.2 Control constructs. Code shall be written using only the five control constructs illustrated in Figures 5 through 9: SEQUENCE, IF-THEN-ELSE, DO-WHILE, DO-UNTIL, CASE. These control constructs refer to the control logic within a Unit while it is executing and do not preclude the calling or passing of processor control to other Units (e.g., subroutines, procedures, functions, exception handlers, interrupt service routines).



CONTROL FLOWS FROM PROCESS A TO THE NEXT IN SEQUENCE, PROCESS B.

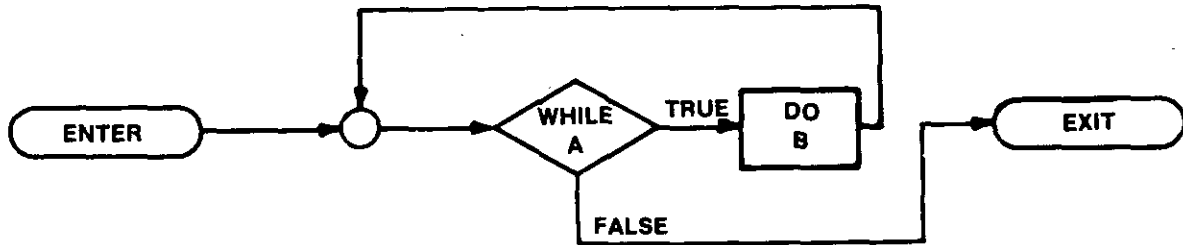
FIGURE 5. Sequence construct.



FLOW OF CONTROL WILL RETURN TO COMMON POINT AFTER EXECUTING PROCESS B OR C. A PREDICATES THE CONDITIONAL EXECUTION.

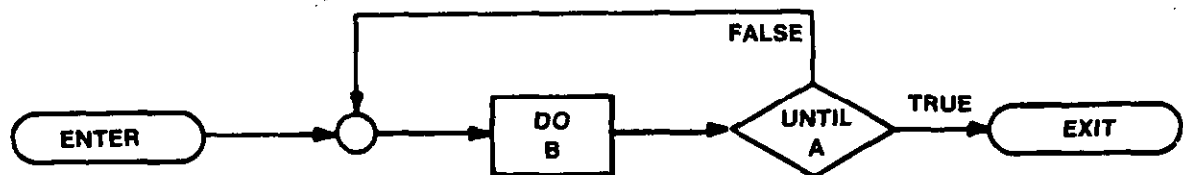
IF OPTION IS TO SKIP A PROCESS PENDING THE CONDITION OF A.

FIGURE 6. IF-THEN-ELSE construct.



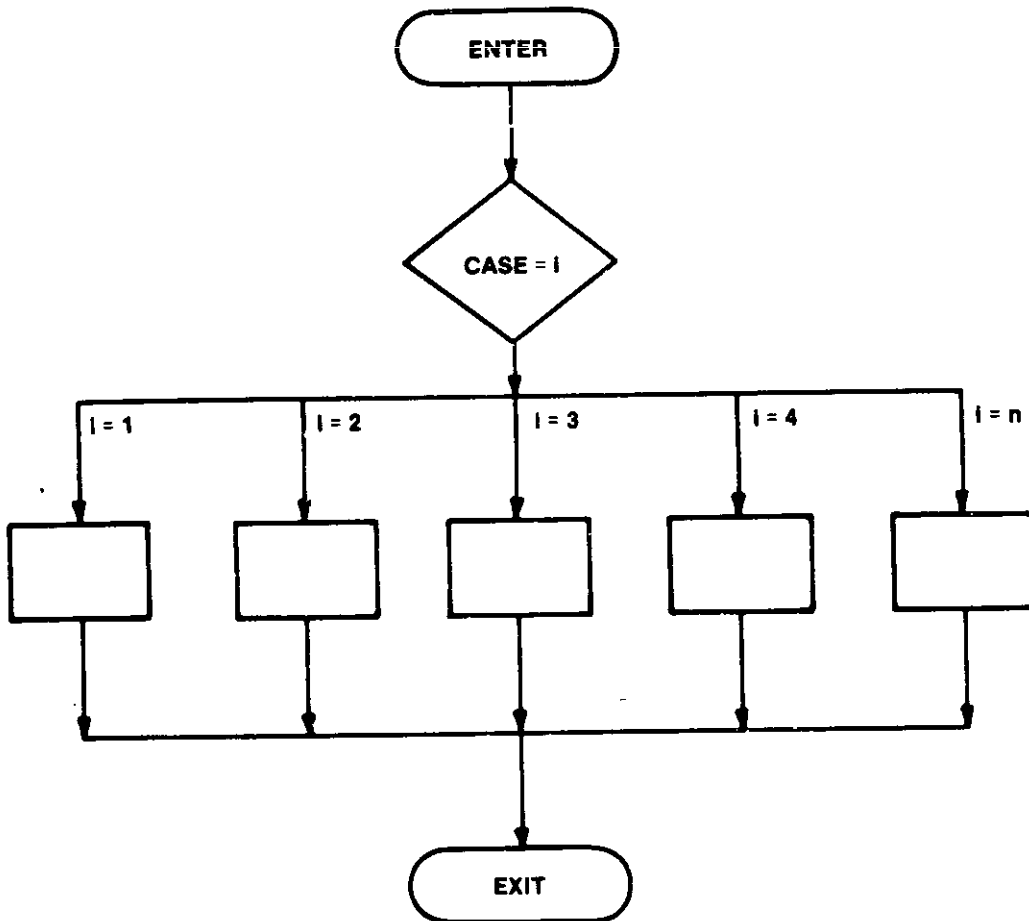
CONDITION A IS EVALUATED. IF FOUND TO BE TRUE, THEN CONTROL IS PASSED TO PROCESS B AND CONDITION A IS (THEN) EVALUATED AGAIN. IF CONDITION A IS FALSE, CONTROL IS PASSED OUT OF THE LOOP.

FIGURE 7. DO-WHILE construct.



SIMILAR TO DO WHILE, EXCEPT THAT THE TEST OF CONDITION A IS PERFORMED AFTER PROCESS B HAS EXECUTED. IF CONDITION A IS TRUE, CONTROL IS PASSED OUT OF THE LOOP.

FIGURE 8. DO-UNTIL construct.



CONTROL IS PASSED TO PROCESS BASED ON THE VALUE OF I.

FIGURE 9. CASE construct.

30.3.3 Modularity. The source code for each Unit shall not exceed, on the average, 100 executable, non-expandable statements or, at most, 200 executable, non-expandable statements. Additionally, Units shall exhibit the following characteristics:

- a. Local variables within different Units shall not share the same storage locations.
- b. Each Unit shall perform a single function.
- c. Modification of a Unit's code during Unit execution shall be prohibited.
- d. Each Unit shall be uniquely named.
- e. All Units shall follow a standard format consisting of prologue, declarative statements, and executable statements or comments, in that order.
- f. Except for error exits, each Unit shall have a single entry point and a single exit point.
- g. Coding style conventions shall be consistent among all Units.

30.3.4 Symbolic parameters. To the maximum extent practical, symbolic parameters shall be used, in lieu of specific numeric values, to represent constants, relative location within a table, and size of data structure.

30.3.5 Naming. Naming conventions shall be uniform throughout the CSCI and shall employ meaningful names which clearly identify the constant, variable, function performed, and any other objects used in the CSCI, to a reader of the source code. Language keywords shall not be used as identifiers.

30.3.6 Mixed mode operations. Mixed mode operations shall be avoided (e.g., arithmetic between real numbers and integer numbers). However, if it is necessary to use them, they shall be clearly identified and described using prominent comments within the source code.

30.3.7 Paragraphing, blocking, and indenting. Paragraphing, blocking by blank lines, and indenting shall be used to enhance the readability of the code.

30.3.8 Complicated expressions. Compound negative Boolean expressions shall be prohibited. Nesting beyond five levels should be avoided.

30.3.9 Compound expressions. The order of evaluation for compound expressions shall be clarified through the use of parentheses and spacing.

30.3.10 Single statement. Each line of source code shall contain, at most, one executable statement.

30.3.11 Comments. Comments shall be set off from the executable source code in a uniform manner. Before each Unit's executable section, a prologue section shall describe the following details:

- a. The Unit's purpose and how it works.
- b. Functions, performance requirements, and external interfaces of the CSCI that the Unit helps implement.
- c. Other Units (subroutines, procedures, functions) called and the calling sequence.
- d. Inputs and outputs, including data files referenced during Unit entry or execution. For each referenced file, the name of the file, usage (input, output, or both), and brief summary of the purpose for referencing the file.
- e. Use of global and local variables and, if applicable, registers and memory locations.
- f. The identification of special tasks that are internally defined, and the size/structure of which are based on external requirements.
- g. The programming department or section responsible for the Unit.
- h. Date of creation of the Unit.
- i. Date of latest revision, revision number, problem report number and title associated with revision.

30.3.12 Error and diagnostic messages. To the maximum extent practical, all error and diagnostic messages shall be presented in a uniform manner and shall be self-explanatory. They shall not require the operator to perform table look-ups or further processing of any kind to interpret the message.

GUIDELINES FOR TAILORING THIS STANDARD

40. General.

40.1 Scope. This appendix provides guidance for the cost-effective tailoring of the requirements of this standard for the development and acquisition of Mission-Critical Computer System software. This appendix serves as guidance for the agency responsible for the preparation of contract requirements and does not form a part of the contract. This appendix provides guidance for the tailoring of software requirements allocated from a System/Segment Specification (SSS). In cases where the software requirements are allocated from a Prime Item Development Specification (PIDS) or Critical Item Development Specification (CIDS), the guidance provided in the Tailoring Handbook* should be considered.

40.2 Purpose. The guidelines contained herein aid in implementing the Department of Defense Directive 4120.21, Specification and Standards Application, which requires all DOD components to apply selectively and to tailor military specifications and standards prior to their contractual imposition. This appendix provides guidelines for tailoring development requirements for Mission-Critical Computer System software. These guidelines help accommodate variations in:

- a. The software development processes used during the system life cycle.
- b. Software characteristics and intended end use.
- c. Acquisition strategies and project management styles for software development.

40.3 Objective. The guidelines in this appendix address the following tailoring objectives:

- a. Eliminating inapplicable and unnecessary requirements.
- b. Eliminating redundancy and inconsistency with other contract specifications and standards.
- c. Promote the use of commercial and reusable software.

40.4 Tailoring Approach. The above tailoring objectives are achieved by a four step-tailoring process:

* Planned for future release.

- a. Step 1 - Classify the required software by categories.
- b. Step 2 - Select applicable contract data items.
- c. Step 3 - Tailor the activities, products, and reviews required during each software development phase.
- d. Step 4 - Tailor the requirements for the selected data items.

40.5 Tailoring Considerations.

40.5.1 Relationship to the Statement of Work (SOW) and the Contract Data Requirements List (CDRL). A typical contract is defined by documents that include the following, or their equivalent:

- a. A statement of work identifying tasks to be accomplished (SOW)
- b. A schedule of contract line items, articles, services, or some combination to be delivered (schedule)
- c. A list of data items, including format, content, and delivery requirements (CDRL)
- d. A specification of the characteristics for the contract line items, articles, and services (Specification).

This software development standard is invoked as a work task by a citation in the SOW. Tailoring of this standard for each of the software categories is accomplished by including appropriate SOW statements which enumerate the changes. Selection of deliverable software documentation for each software category is invoked by citations in the CDRL. These citations cite the appropriate DID to invoke the format and content of the documentation. Tailoring of the DIDs for each software category is accomplished by including appropriate statements in the CDRL enumerating the changes. Such changes could include incorporating the requirements of several DIDs into a single document and thereby eliminate the need for separate DIDs (e.g. incorporate the SCMP, SQEP, and SSPM into the SDP).

40.5.2 Offeror participation in tailoring. Cost-effective tailoring requires that this standard and its related DIDs be tailored to the project requirements and the unique characteristics of the software. The contracting agency is ultimately responsible for this effort. However, the offeror should be given an opportunity to recommend changes and to identify requirements considered appropriate. The contracting agency should request, in the instructions for proposal

preparation, that the offeror recommend the tailoring details in the proposal. The tailoring process should be finalized prior to contract award.

40.6 Tailoring Process

40.6.1 Step 1 - Classify the required software by categories. The software developed for Mission-Critical Computer Systems, and the software used in that development, can be divided into the five categories identified below.

40.6.1.1 Category 1. Deliverable software to be developed and designated as a CSCI. Designating software as a CSCI typically imposes all of the development, documentation, test, review, and control requirements on the software. Some of the factors influencing the decision to designate software as a CSCI are:

- a. Functional complexity
- b. Size
- c. Criticality
- d. Interface complexity
- e. Database complexity
- f. Integration complexity
- g. Complexity of security requirements
- h. Certification requirements
- i. Probability of change
- j. Intended end-use
- k. Support concept
- l. Development location(s)
- m. Schedule.

40.6.1.2 Category 2. Deliverable software to be developed and designated as part of a System or an HWCI. Designating software as part of an HWCI typically imposes fewer requirements than Category 1. Such software may be embedded in firmware devices and may not be expected to undergo significant change. Within the framework of this standard, the contractor may propose the tailoring details applicable to such software, subject to contracting agency approval. Some of the factors influencing the decision to designate software as part of an HWCI are:

- a. Size
- b. Complexity
- c. Probability of change
- d. Intended end-use.

40.6.1.3 Category 3. Non-deliverable software. The controls imposed on non-deliverable software vary widely, depending on the use of the software. Within the framework of this standard, the contractor may propose control provisions for non-deliverable software, subject to contracting agency approval. Some of the factors to consider in establishing control provisions for non-deliverable software are:

- a. Used in formal testing of deliverable products
- b. Used in informal testing
- c. Used to support manufacture of a deliverable item
- d. Used for scientific simulation
- e. Used as an analysis tool in hardware or software design
- f. Probability of change
- g. Duration of use within the software development cycle
- h. Developed software vs. commercially available software.

40.6.1.4 Category 4. Unmodified commercially available and reusable software used in a deliverable CSCI or HWCI. Within the framework of this standard, approval to use unmodified commercially available and reusable software in a deliverable CSCI or HWCI depends on the associated data rights, documentation, and certification evidence which the contractor proposes to provide the contracting agency. Some of the factors to consider in accepting the proposed data rights, documentation, and certification evidence are:

- a. Support plans
- b. Budget constraints
- c. Proprietary information
- d. Duration of project
- e. Product evolution strategy.

40.6.1.5 Category 5. Previously developed software undergoing modifications (such as upgrades, modified GFS, and modified commercially available software). The requirements imposed on modifications to previously developed software vary widely. Some of the factors to consider in establishing the requirements are:

- a. Existing documentation
- b. Available support tools
- c. Modification vs. enhancement
- d. GFS vs. commercial software vs. developed software
- e. Duration of project
- f. Product evolution strategy.

40.6.1.6 Category Summary. These categories reflect the software's past history (if any) and its intended end use. Each category requires a different approach to achieve cost effective management of its software through tailoring the application of this standard and its related DIDs. For this step, it is first necessary to identify each type of software associated with the development program (e.g., operational, diagnostic, and support software). Then, identify how each of these types might consist of software from one or more categories (e.g., operational software includes newly-developed, unmodified reuseable, and some modified GFS components). Then, summarize for each category the different types of software with components within the category. The nature of the software types within any given category will influence the tailoring process for that category.

40.6.2 Step 2 - Select Contract Data Items. The contract data items associated with this standard fall into four categories: Management, Engineering, Test, and Operational and Support. Each of the data items is typically associated with either a system, an individual CSCI, or group of CSCIs. Some of the data items are typically required, while others may be required depending upon project-unique characteristics (see Table I).

40.6.2.1 Management data items. The following data items are in the management category:

- Software Development Plan (SDP)
- Software Configuration Management Plan (SCMP)
- Software Standards and Procedures Manual (SSPM)
- Software Quality Evaluation Plan (SQEP).

TABLE I. Typical data item selection range.

DID TITLE	TYPICALLY REQUIRED	MAY BE COVERED IN ANOTHER DATA ITEM	MAY BE VENDOR-SUPPLIED
SDP	X		
SCMP		X	
SSPM		X	
SQEP		X	
SSS	X		
SRS	X		
IRS		X	
STLDD	X		
SDDD	X		
IDD		X	
DBDD		X	
SPS	X		
VDD	X		
ECP	X		
SCN	X		
STP	X		
STD	X		
STPR	X		
STR	X		
OCD	X		
CSOM			X
SUM			X
CSDM			X
SPM			X
FSM			X
CRISD	X		

The SDP, SCMP, SSPM, and SQEP typically define the contractor's approach to developing all the software in the system, or the software for a group of CSCIs. All the development plans may be described in a single SDP, or broken out into two or more documents (see Figure 10). Some of the factors to consider in selecting the appropriate management documentation are:

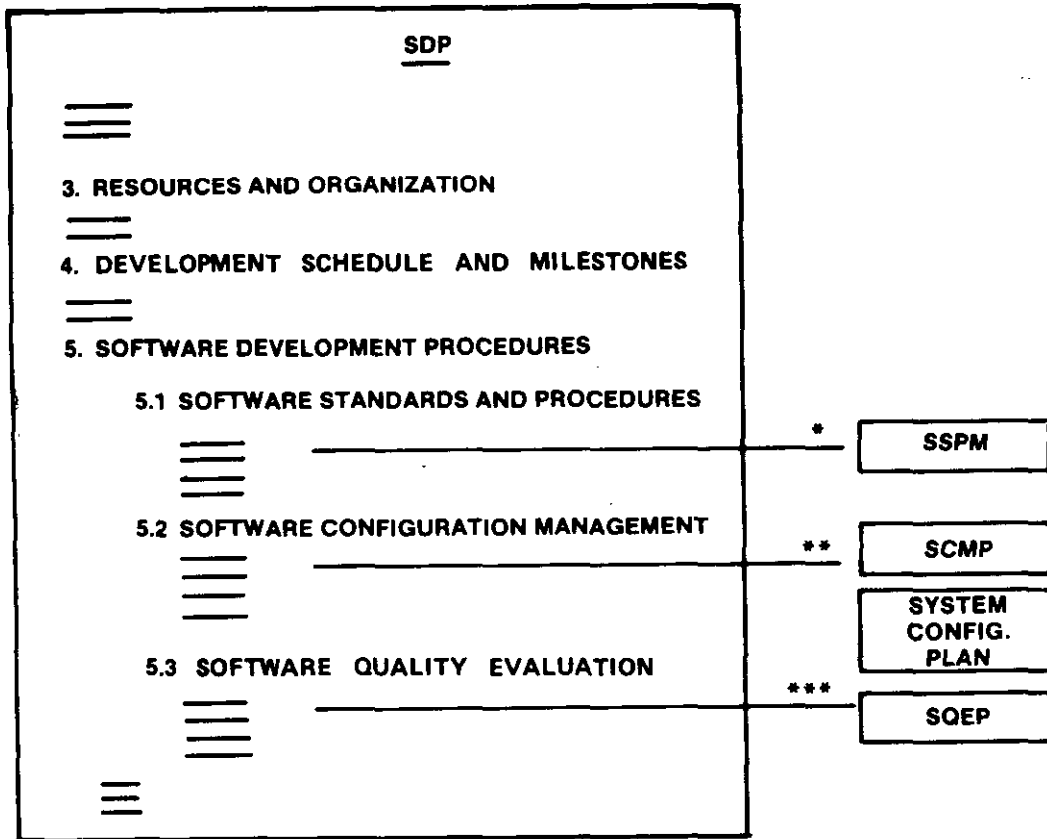
- a. Budget constraints
- b. Multiple contractors or subcontractors
- c. Proprietary information
- d. Project size
- e. Organizational complexity
- f. Complexity of development process
- g. Complexity of development environment
- h. Applicable software categories.

40.6.2.2 Engineering data items. The following data items are in the engineering category:

- System/Segment Specification (SSS)
- Software Requirements Specification (SRS)
- Interface Requirements Specification (IRS)
- Software Top Level Design Document (STLDD)
- Software Detailed Design Document (SDDD)
- Interface Design Document (IDD)
- Data Base Design Document (DBDD)
- Software Product Specification (SPS)
- Version Description Document (VDD)
- Engineering Change Proposal (ECP)
- Specification Change Notice (SCN).

The SSS defines the requirements for the entire system, or segment of the system.

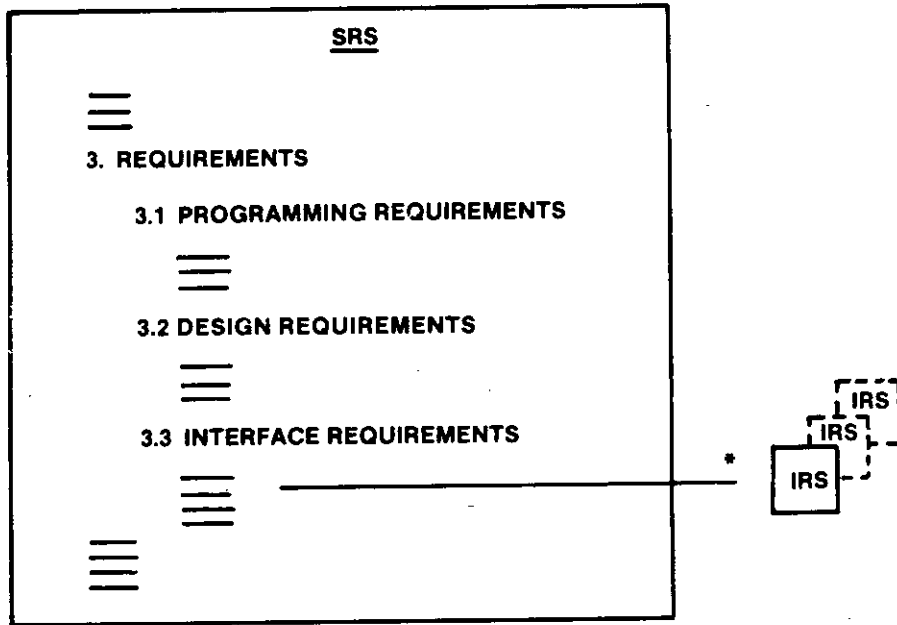
The SRS specifies the requirements for an individual CSCI. The interfaces for each CSCI may be specified in one or more IRSS (see Figure 11). Some of the factors to consider in selecting the appropriate requirements documentation are:



NOTES:

- * SOFTWARE STANDARDS AND PROCEDURES MAY BE PROVIDED IN A SEPARATE SSPM
- ** SOFTWARE CONFIGURATION MANAGEMENT PROCEDURES MAY BE PROVIDED IN A SEPARATE SCMP OR SYSTEM CONFIGURATION MANAGEMENT PLAN
- *** SOFTWARE QUALITY EVALUATION PROCEDURES MAY BE PROVIDED IN A SEPARATE SQEP

FIGURE 10. Relationship among management documents.



NOTES:

* INTERFACE REQUIREMENTS MAY BE SPECIFIED IN ONE OR MORE SEPARATE IRSs.

FIGURE 11. RELATIONSHIP AMONG REQUIREMENTS DOCUMENTS.

DOD-STD-2167
APPENDIX D

- a. Number of interfaces
- b. Number of development groups
- c. Complexity of interfaces
- d. Number of contractors or subcontractors
- e. Applicable software categories.

The STLDD defines the top-level design and the SDDD defines the detailed design for an individual CSCI. The detailed design of the CSCI's data base(s) and external interfaces may be defined in the SDDD or one or more DBDDs and IDDs respectively (see Figure 12). Some of the factors to consider in selecting the appropriate design documentation are:

- a. Interface requirements in separate IRS(s) (separate IDD for each IRS)
- b. Number of data bases
- c. Complexity of data base(s)
- d. Probability of change to data base(s).

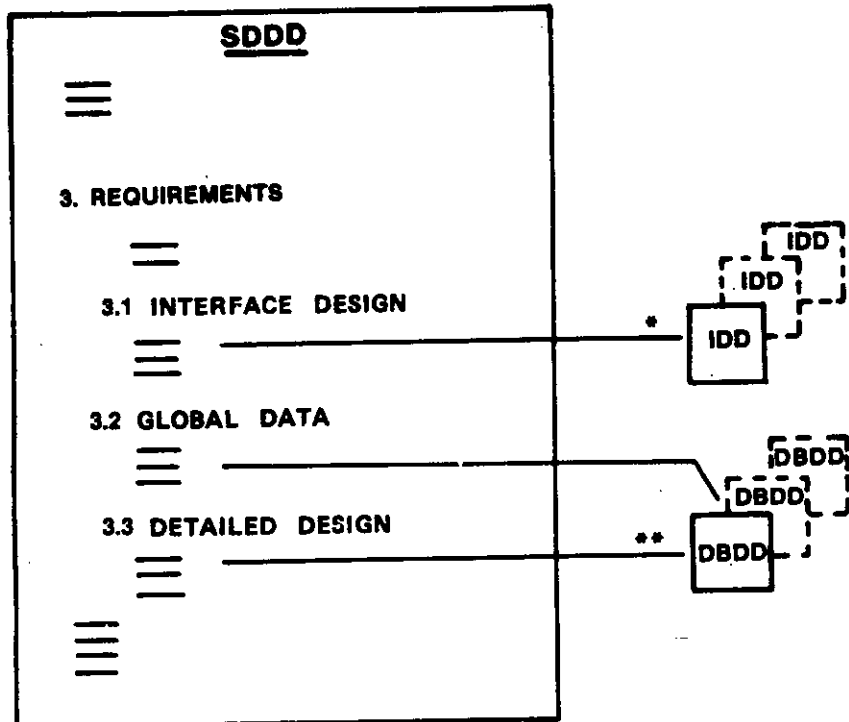
The SPS specifies the "as-built" description of an individual CSCI. The VDD identifies the exact version of an individual CSCI. ECPs and SCNs identify changes to formal baselines.

40.6.2.3 Test data items. The following data items are in the test category:

Software Test Plan (STP)
Software Test Description (STD)
Software Test Procedure (STPR)
Software Test Report (STR).

The test documents identify test information for an individual CSCI. The STP describes the contractor's plans for formal and informal testing. The STD identifies test cases for all formal tests of the CSCI. The STPR describes the step-by-step procedures for executing each formal test. STRs record the results of one or more formal tests.

DOD-STD-2167
APPENDIX D



NOTES:

- * DETAILED DESIGN OF EXTERNAL INTERFACES MAY BE PROVIDED IN ONE OR MORE SEPARATE **IDD**s.
- ** DETAILED DESIGN OF DATA BASE(S) MAY BE PROVIDED IN ONE OR MORE SEPARATE **DBDD**s.

FIGURE 12. Relationship among design documents.

40.6.2.4 Operational and Support data items. The following data items are in the operational and support category:

- Operational Concept Document (OCD)
- Computer System Operator's Manual (CSOM)
- Software User's Manual (SUM)
- Computer System Diagnostic Manual (CSDM)
- Software Programmer's Manual (SPM)
- Firmware Support Manual (FSM)
- Computer Resources Integrated Support Document (CRISD).

The operational documents define the information required to operate the computer system(s) and associated software. The OCD identifies and describes the mission of the system and its operational and support environments. It also describes the functions and characteristics of the computer system within the overall system. The CSOM defines procedures for operating a computer system. The SUM defines procedures to execute one or more CSCIs. The entire SUM, or portions thereof, may be vendor-supplied, if commercially available software is used.

The support documents define the information required to support the computer system and associated software. The CSDM defines procedures to identify and isolate faults in a computer system. The SPM defines the programming aspects of a computer system. The FSM defines procedures to modify or replace firmware devices of a system. The CRISD defines the information required to support all the contractually deliverable software, or a portion thereof.

The CSDM, FSM, SPM, SUM, and CSOM, or portions thereof, may be vendor-supplied and may not be required from the development contractor.

40.6.2.5 Additional Guidance. Additional guidance on the selection of appropriate contract data items may be found in the Tailoring Handbook* related to this standard.

40.6.3 Step 3 - Tailor the activities, products, and reviews. This standard identifies applicable requirements for activities, products, reviews, and baselines/Developmental Configurations for each of the six software development phases identified in 4.1. The products of each phase consist of the contract deliverable data items as well as internal, non-deliverable items. Tailoring the requirements of each phase for each software category is accomplished by deletion of the affected paragraphs in this standard. Detailed tailoring guidance for each software development phase may be found in the Tailoring Handbook* related to this standard.

* Planned for future release.

40.6.4 Step 4 - Tailor the requirements of selected data items.
All of the requirements in the data items selected for the project may not be appropriate. Tailoring the data items is accomplished by deletion of the affected paragraphs in each selected data item for each software category. Detailed guidance for tailoring each data item may be found in the Tailoring Handbook* related to this standard.

* Planned for future release.

Custodians:

Navy - EC
Army - AM
Air Force - 10,26

Preparing Activity:

Navy - EC
(Project MCCR-0005)

Review Activities:

Army - AD,AR,AY,CR,ER,MD,MI
Navy - EC,SH,AS,OM
Air Force - 10,26

INSTRUCTIONS: In a continuing effort to make our standardization documents better, the DoD provides this form for use in submitting comments and suggestions for improvements. All users of military standardization documents are invited to provide suggestions. This form may be detached, folded along the lines indicated, taped along the loose edge (*DO NOT STAPLE*), and mailed. In block 5, be as specific as possible about particular problem areas such as wording which required interpretation, was too rigid, restrictive, loose, ambiguous, or was incompatible, and give proposed wording changes which would alleviate the problems. Enter in block 6 any remarks not related to a specific paragraph of the document. If block 7 is filled out, an acknowledgement will be mailed to you within 30 days to let you know that your comments were received and are being considered.

NOTE: This form may not be used to request copies of documents, nor to request waivers, deviations, or clarification of specification requirements on current contracts. Comments submitted on this form do not constitute or imply authorization to waive any portion of the referenced document(s) or to amend contractual requirements.

(Fold along this line)

(Fold along this line)

DEPARTMENT OF THE NAVY
SPACE AND NAVAL WARFARE SYSTEMS
COMMAND
WASHINGTON, D.C. 20363-5100



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE \$300



POSTAGE WILL BE PAID BY THE DEPARTMENT OF THE NAVY

COMMANDER
SPACE AND NAVAL WARFARE SYSTEMS COMMAND
ATTN: SPAWAR 8111
WASHINGTON, DC 20363-5100



STANDARDIZATION DOCUMENT IMPROVEMENT PROPOSAL
(See Instructions - Reverse Side)

1. DOCUMENT NUMBER		2. DOCUMENT TITLE	
3a. NAME OF SUBMITTING ORGANIZATION		4. TYPE OF ORGANIZATION <i>(Mark one)</i>	
b. ADDRESS <i>(Street, City, State, ZIP Code)</i>		<input type="checkbox"/> VENDOR	
		<input type="checkbox"/> USER	
		<input type="checkbox"/> MANUFACTURER	
		<input type="checkbox"/> OTHER <i>(Specify):</i> _____	
5. PROBLEM AREAS			
a. Paragraph Number and Wording:			
b. Recommended Wording:			
c. Reason/Rationale for Recommendation:			
6. REMARKS			
7a. NAME OF SUBMITTER <i>(Last, First, MI) - Optional</i>		b. WORK TELEPHONE NUMBER <i>(Include Area Code) - Optional</i>	
c. MAILING ADDRESS <i>(Street, City, State, ZIP Code) - Optional</i>		8. DATE OF SUBMISSION <i>(YYMMDD)</i>	