

# MILITARY STANDARD

## DEFENSE SYSTEM SOFTWARE DEVELOPMENT

**Distribution Statement A.** Approved for public release; distribution is unlimited.

---

- [Background and Disclaimer](#)
  - [Approval and Improvements](#)
  - [Foreward](#)
  - Body
    - [1 Scope](#)
    - [2 Referenced Documents](#)
    - [3 Definitions](#)
    - [4 General Requirements](#)
      - [4.1 Software development management](#)
      - [4.2 Software engineering](#)
      - [4.3 Formal qualification testing](#)
      - [4.4 Software product evaluations](#)
      - [4.5 Software configuration management](#)
      - [4.6 Transitioning to software support](#)
    - [5 Detailed Requirements](#)
      - [5.1 System requirements analysis/design](#)
      - [5.2 Software requirements analysis](#)
      - [5.3 Preliminary design](#)
      - [5.4 Detailed design](#)
      - [5.5 Coding and CSU testing](#)
      - [5.6 CSC integration and testing](#)
      - [5.7 CSCI testing](#)
      - [5.8 System integration and testing](#)
    - [6 Notes](#)
  - Appendixes
    - [A. List of acronyms and abbreviations](#)
    - [B. Requirements for software coding standards](#)
    - [C. Category and priority classifications for problem reporting](#)
    - [D. Evaluation Criteria](#)
- 

## Background and Disclaimer

I created this online version of (the (in)famous) DOD-STD-2167A by manually typing it in my word processor from a hardcopy I obtained at work. My reason for this (possibly foolish) act was that since I had to use it, I might as well know it, and forcing every word of it into my eyes and back out my fingers seemed like a good way to imprint it on my brain.

When I first started using the WWW, I thought that this document would be a great contribution to the Web, so I converted the WP5 file to a TXT file, and proceeded to hand insert all the HTML markup tags, and then "published" it by linking it to my home page. I understand that the [Ada WWW pages](#) now reference this document.

Currently in the queue, I have several of the more important DIDs in WP5 form that I will convert and link in, and I am using a drawing program to duplicate the figures, which I will probably convert to .EPS files and also link in. Links to these DIDs and figures exist in this document, but currently point at nonexistent files.

I have no plans of putting the pending replacement for this standard, *DOD-STD-498*, online. If anyone out there wants to do so, be my guest. I really have no idea what the copyright status of this document is, but I strongly assume that it is in the public domain. The conversion of this document to online form is in no way associated with the US Government, the Department of Defense, or my employer(s). I make no guarantees that this document is complete or correct. I make no guarantees that it is "proper" HTML. If you really want a "good" copy, go buy one from the US Government.

Oh, but if you do find any typos or other errors, send me an email and I'll probaby fix it.

Mark Atwood  
[matwood@peruvian.cs.utah.edu](mailto:matwood@peruvian.cs.utah.edu)

---

## Approval and Improvments

1. This Military Standard is approved for use by all Departments and Agencies of the Department of Defense.
2. Beneficial comments (recommendations, additions, deletions) and any pertinent data which may be of use in improving this document should be addressed to:

Commander  
Space and Naval Warfare Systems Command  
ATTN: SPAWAR - 3212  
Washington, DC 20363-5100

by using the self-addressed *Standardization Document Improvement Proposal (DD Form 1426)* appearing at the end of this document or by letter.

---

## Foreword

1. This standard establishes uniform requirements for the software development that are applicable throughout the system life cycle. The requirements of this standard provide the basis for Government insight into a contractor's software development, testing, and evaluation efforts.
2. This standard is not intended to specify or discourage the use of any particular software development method. The contractor is responsible for selecting software development methods (for example, rapid prototyping) that best support the achievement of contract requirements.

3. This standard, together with the other DOD and military document referenced in Section 2, provides the means for establishing, evaluating, and maintaining quality in software and associated documentation.
  4. Data Item Descriptions (DIDs) applicable to this standard are listed in Section 6. These DIDs describe a set of documents for recording the information required by this standard. Production of deliverable data using automated techniques is encouraged.
  5. Per *DODD 5000.43, Acquisition Streamlining*, this standard must be appropriately tailored by the program manager to ensure that only cost-effective requirements are cited in defense solicitations and contracts. Tailoring guidance can be found in *DOD-HDBK-248, Guide for Application and Tailoring of Requirements for Defense Material Acquisitions*.
- 

## 1 Scope

### 1.1 Purpose.

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

### 1.2 Application.

The requirements of this standard apply to the development of Computer Software Configuration Items (CSCIs). This standard applies to the extent specified in the contract clauses, the Statement of Work (SOW), and the Contract Data Requirements List (CDRL).

#### 1.2.1 System Development.

This standard should be used in conjunction with *MIL-STD-499, Engineering Management*, for total system development.

#### 1.2.2 Firmware.

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

#### 1.2.3 Software developed by Government agencies.

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

#### 1.2.4 Other applications.

While the requirements of this standard apply to Computer Software Configuration Items (CSCIs), these requirements may be selectively applied to the development of software not identified as a CSCI (such as, software portions of hardware configuration items and firmware, and non-deliverable software). In such cases, the term CSCI may be interpreted to refer to the selected software.

### 1.3 Tailoring of this standard.

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

---

## 2 Referenced Documents

### 2.1 Government documents.

#### 2.1.1 Specifications, standards, and handbooks.

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

#### MILITARY STANDARDS

DOD-STD-480

Configuration Control - Engineering Changes, Deviations, and Waivers

MIL-STD-481

Configuration Control - Engineering Changes, Deviations, and Waivers (Short Form).

MIL-STD-490

Specification Practices

MIL-STD-499

Engineering Management

MIL-STD-1521

Technical Reviews and Audits for Systems, Equipments, and Computer Software

#### 2.1.2 Other Government documents, drawings, and publications.

None.

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

### 2.2 Other publications.

None.

### 2.3 Order of precedence.

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

---

## 3 Definitions

### **3.1 Allocated Baseline.**

See *DOD-STD-480*.

### **3.2 Authentication.**

Determination by the Government that specification content is acceptable.

### **3.3 Baseline.**

See *DOD-STD-480*.

### **3.4 Computer data definition.**

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

### **3.5 Computer hardware.**

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

### **3.6 Computer resource.**

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

### **3.7 Computer software (or software).**

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

### **3.8 Computer Software Component (CSC).**

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

### **3.9 Computer Software Configuration Item (CSCI).**

A configuration item for computer software.

### **3.10 Computer software documentation.**

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

### **3.11 Computer Software Unit (CSU).**

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

### **3.12 Configuration Identification.**

See *DOD-STD-480*.

### **3.13 Configuration Item.**

See *DOD-STD-480*.

### **3.14 Contracting Agency.**

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

### **3.15 Developmental Configuration.**

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

### **3.16 Evaluation.**

The process of determining whether an item or activity meets specified criteria.

### **3.17 Firmware.**

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

### **3.18 Formal Qualification Testing (FQT).**

A process that allows the contracting agency to determine whether a configuration item complies with the allocated baseline of that item.

### **3.19 Functional Baseline.**

See *DOD-STD-480*.

### **3.20 Hardware Configuration Item (HWCI).**

A configuration item for hardware.

### **3.21 Independent Verification and Validation (IV&V).**

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

### **3.22 Non-developmental software (NDS).**

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

### **3.23 Product Baseline.**

See *DOD-STD-480*.

### **3.24 Release.**

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

### **3.25 Reusable Software.**

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

### **3.26 Software development file (SDF).**

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

### **3.27 Software development library (SDL).**

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

### **3.28 Software engineering environment.**

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

### **3.29 Software support.**

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

### **3.30 Software test environment.**

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

### **3.31 System Specification.**

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

### **3.32 Validation.**

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

### **3.33 Verification.**

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

### **3.34 Version.**

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

### **3.35 Definition of acronyms used in this standard.**

See Appendix A.

---

## **4 General Requirements**

### **4.1 Software development management.**

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

#### **4.1.1 Software development process.**

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

- System Requirements Analysis/Design

- Software Requirements Analysis
- Preliminary Design
- Detailed Design
- Coding and CSU Testing
- CSC Integration and Testing
- CSCI Testing
- System Integration and Testing

#### **4.1.2 Formal reviews/audits.**

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

#### **4.1.3 Software development planning.**

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

#### **4.1.4 Risk management.**

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

#### **4.1.5 Security.**

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

#### **4.1.6 Subcontractor management.**

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

#### **4.1.7 Interface with the software IV&V agent.**

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

#### **4.1.8 Software development library.**

The contractor shall establish a software development library (SDL). The contractor shall document and implement procedures for controlling software and associated documentation residing within the SDL. The

contractor shall maintain the SDL for the duration of the contract.

#### **4.1.9 Corrective action process.**

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

- The process shall be closed loop, ensuring that all detected problems are promptly reported and entered into the corrective action process, action is initiated on them, resolution is achieved, status is tracked and reported, and records of the problems are maintained for the life of the contract.
- Inputs to the corrective action process shall consist of problem/change reports and other discrepancy reports.
- Each problem shall be classified by category and by priority. The categories and priorities identified in [Appendix C](#) shall be included in the category and priority classifications.
- Analysis shall be performed to detect trends in the problems reported.

Corrective actions shall be evaluated to:

- verify that problems have been resolved, adverse trends have been reversed, and changes have been correctly implemented in the appropriate processes and products, and
- to determine whether additional problems have been introduced.

#### **4.1.10 Problem/change report.**

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

### **4.2 Software engineering.**

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

#### **4.2.1 Software development methods.**

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

#### **4.2.2 Software engineering environment.**

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

#### **4.2.3 Safety analysis.**

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

#### **4.2.4 Non-developmental software.**

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

#### **4.2.5 Computer software organization.**

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

#### **4.2.6 Traceability of requirements to design.**

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

#### **4.2.7 High order language.**

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

#### **4.2.8 Design and coding standards.**

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

#### **4.2.9 Software development files.**

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

- Design considerations and constraints
- Design documentation and data
- Schedule and status information
- Test requirements and responsibilities
- Test cases, procedures, and results

#### **4.2.10 Processing resource and reserve capacity.**

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

#### **4.3 Formal qualification testing (FQT).**

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

##### **4.3.1 Formal Qualification Test Planning.**

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

##### **4.3.2 Software test environment.**

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

##### **4.3.3 Independence in FQT activities.**

The organizations, functions, or persons responsible for fulfilling the FQT requirements of this standard shall have the resources, responsibility, authority, and freedom to ensure objective testing and to cause the initiation and verification of corrective action. The persons conducting FQT activities shall not be the persons who developed the software or are responsible for the software. This does not preclude members of the software engineering team from participating in FQT activities. Responsibility for the fulfillment of the FQT requirements shall be assigned and specified in the Software Development Plan (SDP).

#### **4.3.4 Traceability of requirements to test cases.**

The contractor shall document the traceability of the requirements in the Software Requirements Specification (SRS) and Interface Requirements Specification (IRS) that are satisfied or partially satisfied by each test case identified in the Software Test Description (STD). The contractor shall document this traceability in the STD for each CSCI.

#### **4.4 Software product evaluations.**

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

##### **4.4.1 Independence in product evaluation activities.**

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

##### **4.4.2 Final evaluations.**

Prior to submitting each deliverable item to the contracting agency, the contractor shall internally coordinate the item with appropriate organizations for a final evaluation. The objective of each final evaluation shall be to ensure that the deliverable item is acceptable in terms of its ability to satisfy its requirements.

##### **4.4.3 Software evaluation records.**

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

##### **4.4.4 Evaluation criteria.**

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

#### **4.5 Software configuration management.**

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

##### **4.5.1 Configuration identification.**

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

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

#### **4.5.2 Configuration control.**

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

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

#### **4.5.3 Configuration status accounting.**

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

- Provide traceability of changes to controlled products.
- Serve as a basis for communicating the status of configuration identification and associated software.
- Serve as a vehicle for ensuring that delivered documents describe and represent the associated software.

#### **4.5.4 Storage, handling, and delivery of product media.**

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

#### **4.5.5 Engineering Change Proposals.**

The contractor shall prepare Engineering Change Proposals (ECPs) in accordance with *DOD-STD-480* or *MIL-STD-481* as specified in the contract. The contractor shall prepare Specification Change Notices (SCNs) in accordance with *MIL-STD-490*.

## **4.6 Transitioning to software support.**

The contractor shall provide transition support in compliance with the following requirements.

### **4.6.1 Regenerable and maintainable code.**

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

### **4.6.2 Transition planning.**

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

### **4.6.3 Software transition and continuing support.**

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

### **4.6.4 Software support and operational documentation.**

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

- Computer Resource Integrated Support Document (CRISD)
  - Computer System Operator's Manual (CSOM)
  - Software User's Manual (SUM)
  - Software Programmer's Manual (SPM)
  - Firmware Support Manual (FSM)
- 

## **5 Detailed Requirements**

### **5.1 System requirements analysis/design.**

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

#### **5.1.1 Software development management.**

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

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

#### **5.1.2 Software engineering.**

5.1.2.1 The contractor shall analyze the preliminary system specification and shall determine whether the

software requirements are consistent and complete.

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

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

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

### **5.1.3 Formal qualification testing.**

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

### **5.1.4 Software product evaluations.**

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

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

### **5.1.5 Configuration management.**

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

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

## **5.2 Software requirements analysis.**

The contractor shall perform the following software requirements analysis activities.

### **5.2.1 Software development management.**

The contractor shall conduct one or more Software Specification Review(s) (SSR) in accordance with *MIL-STD-1521*. Following successful completion of an SSR and when authenticated by the contracting

agency, the Software Requirements Specifications (SRSs) and associated Interface Requirements Specification (IRS) will establish the Allocated Baseline for the CSCIs.

### **5.2.2 Software engineering.**

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

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

### **5.2.3 Formal qualification testing.**

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

### **5.2.4 Software product evaluations.**

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

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

### **5.2.5 Configuration management.**

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

## **5.3 Preliminary design.**

The contractor shall perform the following preliminary design activities.

### **5.3.1 Software development management.**

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

### **5.3.2 Software engineering.**

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

**5.3.2.2** The contractor shall develop a preliminary design for the interfaces external to each CSCI

documented in the Interface Requirements Specification (IRS). The contractor shall document this information in a preliminary Interface Design Document (IDD).

**5.3.2.3** The contractor shall document in Section 8 of the Software Design Document (SDD) for each CSCI additional engineering information generated during the preliminary design process that is essential to understand the design. The engineering information may include rationale, results of analyses and trade-off studies, and other information that aids in understanding the preliminary design.

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

### **5.3.3 Formal qualification testing.**

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

### **5.3.4 Software product evaluations.**

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

- The Software Design Document (SDD) for each CSCI.
- The preliminary Interface Design Document (IDD)
- The Software Test Plan (STP)
- The CSC test requirements.

### **5.3.5 Configuration management.**

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

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

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

## **5.4 Detailed design.**

The contractor shall perform the following detailed design activities.

### **5.4.1 Software development management.**

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

## 5.4.2 Software engineering.

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

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

**5.4.2.3** The contractor shall document in Section 8 of the Software Design Document (SDD) for each CSCI additional engineering information generated during the detailed design process that is essential to understand the design. The engineering information may include rationale, results of analyses and trade-off studies, and other information that aids in understanding the detailed design.

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

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

## 5.4.3 Formal qualification testing.

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

## 5.4.4 Software product evaluations.

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

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

## 5.4.5 Configuration management.

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

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

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

## **5.5 Coding and CSU testing.**

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

### **5.5.1 Software development management.**

No additional requirements.

### **5.5.2 Software engineering.**

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

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

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

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

### **5.5.3 Formal qualification testing.**

No additional requirements.

### **5.5.4 Software product evaluations.**

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

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

### **5.5.5 Configuration management.**

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

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

configuration control.

## **5.6 CSC integration and testing.**

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

### **5.6.1 Software development management.**

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

### **5.6.2 Software engineering.**

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

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

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

### **5.6.3 Formal qualification testing.**

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

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

### **5.6.4 Software product evaluations.**

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

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

### **5.6.5 Configuration management.**

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

## **5.7 CSCI testing.**

The contractor shall perform the following CSCI testing activities.

### **5.7.1 Software development management.**

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

### **5.7.2 Software engineering.**

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

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

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

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

### **5.7.3 Formal qualification testing.**

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

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

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

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

- The Software Test Report (STR) for each CSCI
- The updated source code and design documentation.

### **5.7.5 Configuration management.**

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

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

## **5.8 System integration and testing.**

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

### **5.8.1 Software development management.**

The contractor shall support the Functional and Physical Configuration Audits (see 5.7.1).

### **5.8.2 Software engineering.**

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

### **5.8.3 Formal qualification testing.**

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

**5.8.3.2** The contractor shall support system integration and testing activities.

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

### **5.8.4 Software product evaluations.**

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

### **5.8.5 Configuration management.**

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

---

## **6 Notes**

### **6.1 Intended use.**

This standard is intended for software development as contracted for by the Department of Defense. The requirements of this standard are written to apply to the development of Computer Software Configuration Items (CSCIs). When software to be developed has not been identified in terms of a CSCI (such as, software portions of hardware configuration items and firmware, and non-deliverable software), the term CSCI may be interpreted to refer to that software and the standard will be applied accordingly.

## 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 27.475-1* 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.

| Paragraph No.                                                                          | Data Requirements Title                                       | Applicable DID No. |
|----------------------------------------------------------------------------------------|---------------------------------------------------------------|--------------------|
| <u>5.1.2.2</u>                                                                         | <u>System/Segment Design Document (SSDD)</u>                  | DI-CMAN-80534      |
| <u>4.1.3</u><br><u>4.3.3</u><br><u>4.4.1</u>                                           | <u>Software Development Plan (SDP)</u>                        | DI-MCCR-80030      |
| <u>4.2.10</u><br><u>5.1.2.3</u><br><u>5.1.3</u><br><u>5.2.2.1</u><br><u>5.2.3</u>      | <u>Software Requirements Specification (SRS)</u>              | DI-MCCR-80025      |
| <u>5.1.2.4</u><br><u>5.2.2.2</u>                                                       | <u>Interface Requirements Specification (IRS)</u>             | DI-MCCR-80026      |
| <u>5.3.2.2</u><br><u>5.4.2.2</u><br><u>5.7.2.2</u>                                     | <u>Interface Design Document (IDD)</u>                        | DI-MCCR-80027      |
| <u>5.3.2.1</u><br><u>5.3.2.3</u><br><u>5.4.2.1</u><br><u>5.4.2.3</u><br><u>5.7.2.1</u> | <u>Software Design Document (SDD)</u>                         | DI-MCCR-80012      |
| <u>4.2.10</u><br><u>5.7.2.4</u>                                                        | <u>Software Product Specification (SPS)</u>                   | DI-MCCR-80029      |
| <u>5.7.5.1</u>                                                                         | <u>Version Description Document (VDD)</u>                     | DI-MCCR-80013      |
| <u>4.3.1</u><br><u>5.3.3</u>                                                           | <u>Software Test Plan (STP)</u>                               | DI-MCCR-80014      |
| <u>4.3.4</u><br><u>5.4.3</u><br><u>5.6.3.1</u><br><u>5.7.3.3</u>                       | <u>Software Test Description (STD)</u>                        | DI-MCCR-80015      |
| <u>5.7.3.2</u>                                                                         | <u>Software Test Report (STR)</u>                             | DI-MCCR-80017      |
| <u>4.6.4</u>                                                                           | <u>Computer System Operator's Manual (CSOM)</u>               | DI-MCCR-80018      |
| <u>4.6.4</u>                                                                           | <u>Software User's Manual (SUM)</u>                           | DI-MCCR-80019      |
| <u>4.6.4</u>                                                                           | <u>Software Programmer's Manual (SPM)</u>                     | DI-MCCR-80021      |
| <u>4.6.4</u>                                                                           | <u>Firmware Support Manual (FSM)</u>                          | DI-MCCR-80022      |
| <u>4.6.2</u><br><u>4.6.4</u>                                                           | <u>Computer Resources Integrated Support Document (CRISD)</u> | DI-MCCR-80024      |

|              |                                   |           |
|--------------|-----------------------------------|-----------|
| <u>4.5.5</u> | Engineering Change Proposal (ECP) | DI-E-3128 |
| <u>4.5.5</u> | Specification Change Notice (SCN) | DI-E-3134 |

(Data item descriptions related to this standard, and identified in section 6 will be approved and listed as such in *DOD 5010.12-L, AMSDL*. Copies of data item descriptions required by the contractors in connection with specified acquisition functions should be obtained from the Naval Publications and Forms Center or as directed by the contracting officer.)

### **6.3 Cost/schedule reporting.**

Contractor cost/schedule reports should be prepared at the CSCI level. The cost reports should indicate budgeted versus actual expenditures and should conform to the Work Breakdown Structure (WBS) applicable to the development effort. These reports should also indicate to the contracting agency planned, actual, and predicted progress.

### **6.4 Subject term (key word) listing.**

- Acquisition
- Baselines
- Code
- Coding and CSU Testing
- Computer
- Computer resources
- Computer software
- Computer software component
- Computer software configuration item
- Computer software unit
- Configuration item
- Configuration management
- CSC
- CSC integration and testing
- CSCI
- CSCI testing
- CSU
- Data item descriptions
- Detailed design
- Developmental configuration
- Engineering environment
- Firmware
- Formal qualification testing
- Non-deliverable software
- Preliminary design
- Qualification
- Requirements analysis
- Risk management
- Safety management
- Software
- Software development
- Software development file
- Software development library

- Software engineering
- Software product evaluation
- Software requirements analysis
- Software support
- System integration and testing
- Test environment
- Testing

## 6.5 Changes from previous issue.

Asterisks or vertical lines are not used in this revision to identify changes with respect to the previous issue due to the extensiveness of the changes.

---

# Appendix A. List of Acronyms and Abbreviations

## 10.1 Purpose.

This appendix provides a list of all acronyms and abbreviations used in this standard, with the associated meaning. This appendix is not a mandatory part of the standard. The material contained in this appendix is for information only.

## 10.2 Acronyms.

|        |                                                             |
|--------|-------------------------------------------------------------|
| CDR    | Critical Design Review                                      |
| CDRL   | Contract Data Requirements List                             |
| CIDS   | Critical Item Development Specification                     |
| CRISD  | Computer Resources Integrated Support Document              |
| CSC    | Computer Software Component                                 |
| CSCI   | Computer Software Configuration Item                        |
| CSOM   | Computer System Operator's Manual                           |
| CSU    | Computer Software Unit                                      |
| DID    | Data Item Description                                       |
| DOD    | Department of Defense                                       |
| DODISS | Department of Defense Index of Specifications and Standards |
| ECP    | Engineering Change Proposal                                 |
| FAR    |                                                             |

|      |                                          |
|------|------------------------------------------|
|      | Federal Acquisition Regulation           |
| FCA  |                                          |
|      | Functional Configuration Audit           |
| FSM  |                                          |
|      | Firmware Support Manual                  |
| FQT  |                                          |
|      | Formal Qualification Testing             |
| GFS  |                                          |
|      | Government Furnished Software            |
| HOL  |                                          |
|      | High Order Language                      |
| HWCI |                                          |
|      | Hardware Configuration Item              |
| IDD  |                                          |
|      | Interface Design Document                |
| I/O  |                                          |
|      | Input/Output                             |
| IRS  |                                          |
|      | Interface Requirements Specification     |
| IV&V |                                          |
|      | Independence Verification and Validation |
| NDS  |                                          |
|      | Non-developmental Software               |
| PCA  |                                          |
|      | Physical Configuration Audit             |
| PDR  |                                          |
|      | Preliminary Design Review                |
| PIDS |                                          |
|      | Prime Item Development Specification     |
| SCN  |                                          |
|      | Specification Change Notice              |
| SDD  |                                          |
|      | Software Design Document                 |
| SDF  |                                          |
|      | Software Development File                |
| SDL  |                                          |
|      | Software Development Library             |
| SDP  |                                          |
|      | Software Development Plan                |
| SDR  |                                          |
|      | System Design Review                     |
| SOW  |                                          |
|      | Statement of Work                        |
| SPM  |                                          |
|      | Software Programmer's Manual             |
| SPS  |                                          |
|      | Software Product Specification           |
| SRR  |                                          |
|      | System Requirements Review               |
| SRS  |                                          |
|      | Software Requirements Specification      |

|      |                                |
|------|--------------------------------|
| SSDD | System/Segment Design Document |
| SSR  | Software Specification Review  |
| SSS  | System/Segment Specification   |
| STD  | Software Test Description      |
| STP  | Software Test Plan             |
| STR  | Software Test Report           |
| SUM  | Software User's Manual         |
| TRR  | Test Readiness Review          |
| VDD  | Version Description Document   |
| WBS  | Work Breakdown Structure       |

---

## **Appendix B. Requirements for Software Coding Standards**

### **10.1 Purpose.**

The purpose of this appendix is to specify language independent requirements for software coding standards. The requirements specified in this appendix are a mandatory part of this standard.

### **10.2 Applicability.**

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

### **10.3 Rules and Conventions.**

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

#### **10.3.1 Presentation Style.**

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

- Indentation and spacing
- The use of capitalization
- Uniform presentation of information throughout the source code (e.g., the grouping together of all the data declarations)

- Use of headers
- Layout of source code listings
- Conditions under which comments are provided and the format to be used
- Size of code aggregates (e.g. on the average 100 or at most 200 executable non-expandable statements).

### 10.3.2 Naming.

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

### 10.3.3 Restrictions on the implementation language.

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

### 10.3.4 Use of language constructs and features.

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

### 10.3.5 Complexity.

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

---

## Appendix C. Category and Priority Classifications for Problem Reporting

### 10.1 Purpose.

This appendix contains requirements for a category and priority classification scheme to be applied to all problems detected in the deliverable software or its documentation that has been placed under contractor configuration control. The requirements specified in this appendix are a mandatory part of this standard.

### 10.2 Classification by category.

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

- **Software problem.** The software does not operate according to supporting documentation and the documentation is correct.
- **Documentation problem.** The software does not operate according to supporting documentation but the software operation is correct.

- **Design problem.** The software operated according to supporting documentation but a design deficiency exists. The design deficiency may not always result in a directly observable operational symptom but possesses the potential for causing further problems.

### 10.3 Classification by priority.

Problems detected in the software or its documentation shall be classified by priority as follows:

- **Priority 1.** A software problem that does one of the following:
    - Prevents the accomplishment of an operational or mission essential capability specified by baselined requirements
    - Prevents the operator's accomplishment of an operational or mission essential capability
    - Jeopardizes personnel safety.
  - **Priority 2.** A software problem that does one of the following:
    - Adversely affects the accomplishment of an operational or mission essential capability specified by baselined requirements so as to degrade performance and for which no alternative work-around solution is known
    - Adversely affects the operator's accomplishment of an operational or mission essential capability specified by baselined requirements so as to degrade performance and for which no alternative work-around solution is known.
  - **Priority 3.** A software problem that does one of the following:
    - Adversely affects the accomplishment of an operational or mission essential capability specified by baselined requirements so as to degrade performance and for which an alternative work-around solution is known
    - Adversely affects the operator's accomplishment of an operational or mission essential capability specified by baselined requirements so as to degrade performance and for which an alternative work-around solution is known.
  - **Priority 4.** A software problem that is an operator inconvenience or annoyance and which does not effect a required operational or mission essential capability.
  - **Priority 5.** All other errors.
- 

## Appendix D. Evaluation Criteria

### 10.1 Purpose.

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

### 10.2 Criteria definitions.

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

#### 10.2.1 Internal consistency.

Internal consistency as used in this standard means that:

- no two statements in a document contradict one another,
- a given term, acronym, or abbreviation means the same thing throughout the document, and
- a given item or concept is referred to by the same name or description throughout the document.

### **10.2.2 Understandability.**

Understandability, as used in this standard means that:

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

### **10.2.3 Traceability to indicated documents.**

Traceability as used in this standard means that the document in question is in agreement with a predecessor document to which it has a hierarchical relationship. Traceability has five elements:

- the document in question contains or implements all applicable stipulations of the predecessor document,
- a given term, acronym, or abbreviation means the same thing in the documents,
- a given item or concept is referred to by the same name or description in the documents,
- all material in the successor document has its basis in the predecessor document, that is, no untraceable material has been introduced, and
- the two documents do not contradict one another.

### **10.2.4 Consistency with indicated documents.**

Consistency between documents, as used in this standard, means that two or more documents that are not hierarchically related are free from contradictions with one another. Elements of consistency are:

- no two statements contradict one another,
- a given term, acronym, or abbreviation means the same thing in the documents, and
- a given item or concept is referred to by the same name or description in the documents.

### **10.2.5 Appropriate analysis, design, and coding techniques used.**

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

### **10.2.6 Appropriate allocation of sizing and timing resources.**

This criterion, as used in this standard, means that:

- the amount of memory or time allocated to a given element does not exceed documented constraints applicable to that element, and
- the sum of the allocated amounts for all subordinate elements is within the overall allocation for an item.

### **10.2.7 Adequate test coverage or requirements.**

This criterion, as used in this standard, means that:

- every specified requirement is addressed by at least one test,
- test cases have been selected for both "average" situation and "boundary" situations, such as minimum and maximum values,
- "stress" cases have been selected, such as out-of-bound values, and
- test cases that exercise combinations of different functions are included.

## **10.3 Additional criteria.**

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

### **10.3.1 Adequacy of quality factors.**

This criterion applies to the quality factor requirements in the Software Requirements Specification (SRS). Aspects to be considered are:

- trade-offs between quality factors have been considered and documented, and
- each quality factor is accompanied by a feasible method to evaluate compliance, as required by the SRS DID.

### **10.3.2 Testability of requirements.**

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

### **10.3.3 Consistency between data definition and data use.**

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

### **10.3.4 Adequacy of test cases, test procedures, (test inputs expected results, evaluation criteria).**

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

### **10.3.5 Completeness of testing.**

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

### **10.3.6 Completeness of retesting.**

Retesting consists of repeating a subset of the test cases and test procedures after software corrections have been made to correct problems found in previous testing. Retesting is considered complete if:

- all test cases and test procedures that revealed problems in the previous testing have been repeated, their results have been recorded, and the results have met acceptance criteria, and
  - all test cases and test procedures that revealed no problems during the previous testing, but that test functions that are affected by the corrections, have been repeated, their results have been recorded, and the results have met acceptance criteria.
-