



COMDTINST M5234.4
MAY 29 1996

COMMANDANT INSTRUCTION M5234.4

Subj: COAST GUARD SOFTWARE DEVELOPMENT AND DOCUMENTATION STANDARDS (CG-SDDS)

Ref: (a) Designated Senior Official (DSO) for Information Resource
Management (IRM) COMDTINST 5230.38
(b) Planning Approval for Automated Information Systems (AIS)
COMDTINST 5231.2

1. PURPOSE: This instruction defines software development and documentation standards for automated information systems (AIS), establishes the Coast Guard Software development and Documentation Standards (CG-SDDS) (Enclosure 1), and assigns Commandant (G-S) with oversight authority for ensuring compliance with the CG-SDDS for all Coast Guard software development initiatives. This instruction applies to all units.
2. ACTION: Area and District Commanders, Commanders of Maintenance and Logistics commands, Chiefs of Headquarters offices, Commanding Officers/Commanders of Headquarters Units, Director of the National Pollution Funds Center shall ensure compliance with this instruction.
3. DIRECTIVES AFFECTED: None.
4. BACKGROUND:
 - a. In an environment of continued federal budget reductions the Coast Guard can increase productivity and reduce waste by standardizing data processing systems. Maintaining diverse systems, with conflicting and redundant rules, processes and databases is inefficient and expensive. With the introduction of standard software development methodologies, resources can be more efficiently
 - b. The Coast Guard recognizes the need to establish uniform requirements for software development, documentation and database standardization. These processes must be capable of supporting a repeatable software development process and ensure that quality assurance practices are established and maintained throughout any development effort. In the past, the Coast Guard

has relied primarily on two Department of Defense (DoD) standards for software development: DOD-STD-2167A and DOD-STD-7935A. Former standards.

- (1) **DOD-STD-2167A**, Defense System Software Development. This standard was used for mission critical systems. It defines a software development process and has 16 Data Item Descriptions (DIDs) defining documentation
 - (2) **DOD-STD-7935A**, DoD Automated Information Systems (AIS) Documentation Standards. This standard was used for automated information systems. It defines the format and content of 11 documents and provides guidance for applying these documents.
- c. A revised Standard, MIL-STD-498 was approved in November 1994. A major justification for its approval was the lack of a viable commercial standard that may be used in lieu of the costly development of a military standard. A commercial standard is not anticipated in the foreseeable future. The MIL-STD-498 represents a consolidation of previous standards. The consolidation was possible by combining mission critical systems (MCCR) with those of automated information systems (AIS) throughout DoD, thus eliminating the need to maintain two separate standards
- d. Commandant (G-STC) has tailored MIL-STD-498 for Coast Guard use, resulting in a consolidated standard which supports Coast Guard business systems software development. Commandant (G-STC) shall support this standard until an acceptable commercial standard becomes available

5. DISCUSSION.

- a. Documentation of all aspects of the software development effort is critical for effective software maintenance following product acceptance. Adoption of the Software Development and Documentation Standards provides the Coast Guard with a consistent documentation set for software development efforts. The standard is structured in a manner directly applicable to contracted development efforts. Documents resulting from its use may be cited within a statement of work as deliverables. The following information provides a brief description of the CG-SDDS and its supporting Data Item Descriptions (DIDS). The DIDS contain format and content information relative to individual documents derived from the CG-SDDS.
- b. CG Software Development and Documentation Standards (CG-SDDS) The CG-SDDS supports eight Data Item Descriptions (DIDS). The standard differentiates between planning and engineering activities and the resulting documentation. The key objective of the standard is to eliminate the notion that the Coast Guard must accept a given deliverable in order for planning or engineering work to take place. Under the CG-SDDS, the planning and engineering work takes place as an integral part of the

development process, regardless of which deliverables are ordered. In addition, a series of joint technical reviews, are included to ensure appropriate progress is maintained throughout the development lifecycle.

c. Supporting data Item Descriptions (DIDS):

- (1) **CG Software Plan (CG-SP)**. A single document encompassing the planning for software development, software installation, and preparation for transition to software support. It is used during the planning phase of development
- (2) **CG Software Requirements Document (CG-SRD)**. A single document encompassing the operational concept and requirements for a system, for one or more Computer Software Configuration Items (CSCI), and for the external interfaces of the CSCIs. It is used in the requirements phase of development.
- (3) **CG Software Design Document (CG-SDD)**. A single document specifying the design of a system, including all CSCIs, CSCI-external interfaces, and any database(s) in the system. It is used in the design phase of development.
- (4) **CG Software Test Document (CG-STD)**. A single document which specifies all test plans, test cases, procedures, and test results. It is used in the test phase of development.
- (5) **CG Software User/Operator Manual (CG-SUOM)**. A single document which provides instructions to hands-on users, computer center software operators, preparers and users of software input/output, and operators of computer hardware. It is used in the final delivery/maintenance phase of development.
- (6) **CG Software Support Document (CG-SSD)**. A Single document which baselines the delivered system through a version description, product specification, and support manuals. It is used in the maintenance phase.
- (7) **CG Consolidated Software Development Document (CG-CSDD)**. A single document describing all aspects of a software development project. It is used on smaller, less complex software systems, in lieu of the documents listed above. It is used in all phases of development.
- (8) **CG Progress Report (CG-PR)**. Provides a standard format for the Contractor to report the status of the development effort. It includes a forum with which to identify assigned tasks, costs and existing or potential problem areas. It is used throughout all phases of development.

d. In addition to the CG-SDDS, Enclosure (2) is the Software Development and Documentation Standards Guide. Enclosure (3) is the Guide to Using the Consolidated Software Development Document. These two guides provide an overview of the relationship between the CG-SDDS and its supporting DIDs. The guides also provide a graphical depiction of the functional activities that occur when the CG-SDDS is applied

to a contract. It is recommended that Enclosures (2) and (3) be read first as an introduction to the CG-SDDS.

6. POLICY. The scope of the Coast Guard Software Development and Documentation Standards (CG-SDDS) encompasses software development in support of Coast Guard missions and goals. Software developers shall ensure compliance with this policy for all new AIS development and for those AIS systems undergoing major redesign/modification. Commandant (G-SCC-2) is available to assist software developers with compliance to this instruction.

D. A. POTTER
Director of Command, Control,
Communications, and Computers

- Encl:
- (1) Software Development and Documentation Standards (CG-SDDS) and supporting Data Item Descriptions (DIDS).
 - (2) Software Development and Documentation Standards Guide.
 - (3) Guide to using the Consolidated Software Development Document.

US Coast Guard
Software Development and Documentation Standards
(CG-SDDS)
April 1996
US Coast Guard Headquarters
Software Policy and Standards Division (G-SCC-2)
2100 Second Street SW
Washington, DC 20593-0001

Note: This document is based on MIL-STD-498 which was prepared by the Harmonization Working Group. The US Coast Guard has made modifications to MIL-STD-498 and plans to adopt this revised document as an internal standard (i.e., Coast Guard Software Development and Documentation Standards, CG-SDDS). MIL-STD-498 remains the foundation for the CG-SDDS. The US Coast Guard takes no credit for the Harmonization Working Group's efforts in developing MIL-STD-498.

CONTENTS

| <u>Paragraph</u> | <u>Page</u> |
|--|-------------|
| 1. Scope | 1 |
| 1.1 Purpose | 1 |
| 1.2 Application | 1 |
| 1.2.1 Contract-specific application..... | 1 |
| 1.2.2 Tailoring..... | 1 |
| 1.2.3 Software developed by Government agencies..... | 1 |
| 1.2.4 References to the CDRL..... | 2 |
| 1.2.5 System development..... | 2 |
| 1.2.6 Meaning of "develop" and "define"..... | 2 |
| 1.2.7 Meaning of "record"..... | 2 |
| 2. REFERENCED DOCUMENTS | 3 |
| 2.1 Government documents | 3 |
| 2.1.1 Specifications, standards, and handbooks..... | 3 |
| 2.1.2 Other Government documents, drawings, and publications..... | 3 |
| 2.2 Non-Government publications | 3 |
| 2.3 Order of precedence | 3 |
| 3. DEFINITIONS | 4 |
| 4. GENERAL REQUIREMENTS | 8 |
| 4.1 Software development process | 8 |
| 4.2 General requirements for software engineering | 8 |
| 4.2.1 Software development methods..... | 8 |
| 4.2.2 Reusable software..... | 8 |
| 4.2.2.1 Evaluating reusable software..... | 8 |
| 4.2.2.2 Incorporating reusable software..... | 9 |
| 4.2.3 Safety analysis..... | 9 |
| 4.2.4 Processing resource and reserve capacity..... | 10 |
| 4.2.5 Recording rationale..... | 10 |
| 4.3 General requirements for software testing | 10 |
| 4.3.1 Independence in testing activities..... | 10 |
| 4.3.2 Testing on target computer system..... | 10 |
| 4.3.3 Stress testing..... | 10 |
| 5. DETAILED REQUIREMENTS | 10 |
| 5.1 Project planning | 12 |
| 5.1.1 Software development planning..... | 12 |
| 5.1.2 CSCI and CSCI integration test planning..... | 12 |
| 5.1.3 System test planning..... | 12 |
| 5.1.4 Software support planning..... | 12 |
| 5.1.5 Software installation planning..... | 13 |

CONTENTS - Continued

| <u>Paragraph</u> | <u>Page</u> |
|---|-------------|
| 5.2 Establishing a software development environment..... | 13 |
| 5.2.1 Software engineering environment..... | 13 |
| 5.2.2 Software test environment..... | 13 |
| 5.2.3 Software development library..... | 13 |
| 5.2.4 Software development files..... | 13 |
| 5.2.5 Design and coding standards..... | 14 |
| 5.2.6 Non-deliverable software..... | 14 |
| 5.3 System requirements analysis..... | 14 |
| 5.3.1 Analysis of user input..... | 14 |
| 5.3.2 Operational concept..... | 14 |
| 5.3.3 System requirements..... | 14 |
| 5.4 System design..... | 15 |
| 5.4.1 System behavioral design..... | 15 |
| 5.4.2 System architectural design..... | 15 |
| 5.5 Software requirements analysis..... | 16 |
| 5.5.1 CSCI engineering requirements..... | 16 |
| 5.5.2 CSCI interface requirements..... | 16 |
| 5.6 Software architectural design..... | 16 |
| 5.6.1 CSCI behavioral design..... | 16 |
| 5.6.2 CSCI architectural design..... | 17 |
| 5.6.3 Database logical design..... | 17 |
| 5.7 Software detailed design..... | 17 |
| 5.7.1 CSCI detailed design..... | 17 |
| 5.7.2 CSCI interface design..... | 17 |
| 5.7.3 Database physical design..... | 17 |

| | | |
|------------|--|-----------|
| 5.8 | Coding and unit testing..... | 18 |
| 5.8.1 | Coding software units..... | 18 |
| 5.8.2 | Populating databases..... | 18 |
| 5.8.3 | Preparing for unit testing..... | 18 |
| 5.8.4 | Performing unit testing..... | 18 |
| 5.8.5 | Revision and retesting..... | 18 |
| 5.8.6 | Recording unit test results..... | 19 |
| | | |
| 5.9 | Unit integration and testing..... | 19 |
| 5.9.1 | Preparing for unit integration and testing.... | 19 |
| 5.9.2 | Performing unit integration and testing..... | 19 |
| 5.9.3 | Revision and retesting..... | 19 |
| 5.9.4 | Recording unit integration and test results... | 19 |

CONTENTS - Continued

| <u>Paragraph</u> | <u>Page</u> |
|--|-------------|
| 5.10 CSCI testing | 19 |
| 5.10.1 Preparing CSCI test cases..... | 20 |
| 5.10.2 Preparing CSCI test procedures..... | 20 |
| 5.10.3 Dry run of CSCI test procedures..... | 20 |
| 5.10.4 Performing CSCI testing..... | 20 |
| 5.10.5 Revision and retesting..... | 20 |
| 5.10.6 Analyzing CSCI test results..... | 20 |
| 5.11 CSCI integration and testing | 20 |
| 5.11.1 Preparing CSCI integration and test cases..... | 20 |
| 5.11.2 Preparing CSCI integration and test procedures..... | 21 |
| 5.11.3 Dry run of CSCI integration and test procedures..... | 21 |
| 5.11.4 Performing CSCI integration and testing..... | 21 |
| 5.11.5 Revision and retesting..... | 21 |
| 5.11.6 Analyzing CSCI integration and test results... | 21 |
| 5.12 System testing | 21 |
| 5.12.1 Preparing system test cases..... | 21 |
| 5.12.2 Preparing system test procedures..... | 22 |
| 5.12.3 Dry run of system test procedures..... | 22 |
| 5.12.4 Performing system testing..... | 22 |
| 5.12.5 Revision and retesting..... | 22 |
| 5.12.6 Analyzing system test results..... | 22 |
| 5.13 Final Acceptance Testing | 22 |
| 5.13.1 Preparing final acceptance test cases..... | 22 |
| 5.13.2 Preparing final acceptance test procedures.... | 23 |
| 5.13.3 Dry run of final acceptance test procedures... | 23 |
| 5.13.4 Performing final acceptance testing..... | 23 |
| 5.13.5 Revision and retesting..... | 23 |
| 5.13.6 Analyzing final acceptance test results..... | 23 |
| 5.14 Preparation for software use and support | 23 |
| 5.14.1 Software user manuals..... | 24 |
| 5.14.2 Computer center software operator manuals..... | 24 |
| 5.14.3 Software input/output manuals..... | 24 |
| 5.14.4 Computer system operator manuals..... | 24 |
| 5.14.5 Computer instruction set architecture manuals..... | 24 |
| 5.14.6 Installation at user sites..... | 24 |
| 5.14.7 Transition to designated support site..... | 24 |
| 5.15 Preparation for software delivery | 25 |
| 5.15.1 Preparing executable code for delivery..... | 25 |
| 5.15.2 Preparing source code for delivery..... | 25 |
| 5.15.3 Software product specifications..... | 25 |
| 5.15.4 Version descriptions..... | 25 |

CONTENTS - Continued

| <u>Paragraph</u> | <u>Page</u> |
|---|--------------------|
| 5.15.5 Functional Configuration Audits..... | 25 |
| 5.15.6 Physical Configuration Audits..... | 25 |
| 5.16 Software product evaluations..... | 25 |
| 5.16.1 In-process software product evaluations..... | 26 |
| 5.16.2 Final software product evaluations..... | 26 |
| 5.16.3 Independence in product evaluation activities..... | 26 |
| 5.16.4 Software product evaluation records..... | 26 |
| 5.17 Software configuration management..... | 26 |
| 5.17.1 Contractor control of development products.... | 27 |
| 5.17.1.1 Contractor-internal configuration identification..... | 27 |
| 5.17.1.2 Contractor-internal configuration control..... | 27 |
| 5.17.2 Interface with Government Configuration Management..... | 28 |
| 5.17.2.1 Engineering Change Proposals..... | 28 |
| 5.17.2.2 Configuration status accounting..... | 28 |
| 5.17.3 Storage, handling, and delivery of project media..... | 28 |
| 5.18 Corrective action and process improvement..... | 28 |
| 5.18.1 Problem/change reports..... | 28 |
| 5.18.2 Corrective action and process improvement system..... | 29 |
| 5.19 Joint reviews..... | 29 |
| 5.19.1 Joint technical reviews..... | 30 |
| 5.19.2 Joint management reviews..... | 30 |
| 5.20 Other management activities..... | 31 |
| 5.20.1 Risk management..... | 31 |
| 5.20.2 Security and privacy..... | 31 |
| 5.20.3 Subcontractor management..... | 31 |
| 5.20.4 Interface with software IV&V agents..... | 31 |
| 5.20.5 Interface with associate contractors..... | 31 |
| 5.20.6 Software management indicators..... | 31 |
| 5.21 Software Quality Assurance | |
| 5.21.1 Software quality assurance evaluations..... | 32 |
| 5.21.2 Software quality assurance records..... | 32 |
| 5.21.3 Independence in software quality assurance.... | 32 |
| 6. NOTES..... | 33 |
| 6.1 Intended use..... | 33 |
| 6.2 Ordering software via CLIN..... | 33 |

CONTENTS - Continued

| | | |
|-----|---|----|
| 6.3 | Data requirements list and cross reference..... | 33 |
| 6.4 | Relationship between standard and CDRL..... | 35 |
| 6.5 | Delivery of database/tool contents..... | 35 |
| 6.6 | Cost/schedule reporting..... | 35 |
| 6.7 | Subject term (key word) listing..... | 35 |

APPENDIXES

Appendix

| | | |
|----------|--|------------|
| A | List of acronyms and abbreviations..... | A-1 |
| | A.1 Purpose..... | A-1 |
| | A.2 Acronyms..... | A-1 |
| B | Requirements for software coding standards..... | B-1 |
| | B.1 Purpose..... | B-1 |
| | B.2 Requirements for coding standards..... | B-1 |
| C | Category and priority classifications for problem reporting..... | C-1 |
| | C.1 Purpose..... | C-1 |
| | C.2 Classification by category..... | C-1 |
| | C.3 Classification by priority..... | C-1 |
| D | Software product evaluations..... | D-1 |
| | D.1 Purpose..... | D-1 |
| | D.2 Required evaluations..... | D-1 |
| | D.3 Criteria definitions..... | D-1 |
| E | Candidate joint management reviews..... | E-1 |
| | E.1 Purpose..... | E-1 |
| | E.2 Assumptions..... | E-1 |
| | E.3 Candidate reviews..... | E-1 |
| F | Candidate management indicators..... | F-1 |
| | F.1 Purpose..... | F-1 |
| | F.2 Candidate indicators..... | F-1 |
| G | Applying CG-SDDS to life cycle models..... | G-1 |
| | G.1 Purpose..... | G-1 |
| | G.2 Candidate life cycle models..... | G-1 |
| | G.3 Selecting an appropriate model..... | G-1 |
| | G.4 Planning the builds for an incremental or evolutionary model..... | G-2 |
| | G.5 Applying CG-SDDS to each life cycle mode..... | G-2 |
| H | Guidance on deliverables..... | H-1 |
| | H.1 Purpose..... | H-1 |
| | H.2 Ordering deliverables..... | H-1 |
| | H.3 Scheduling deliverables..... | H-1 |
| | H.4 Format of deliverables..... | H-1 |
| I | Activities and deliverables for reusable software..... | I-1 |

APPENDIXES - Continued

I.1 Purpose.....I-1
I.2 Interpreting CG-SDDS activities for
 reusable software.....I-1
INDEX.....IN-1

1. SCOPE

1.1 Purpose

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

1.2 Application

- 1.2.1 Contract-specific application. This standard provides a comprehensive set of requirements that may be selectively applied to any software being developed for the Government. The standard applies to each item or type of software named in a contract to the extent specified in the contract. For example, the contract may specify that the standard does (or does not) apply to deliverable application software, deliverable training software,

deliverable software in the software engineering and test environments, non-deliverable software in the software engineering and test environments, non-deliverable software used in the manufacture of hardware items, or other items or types of software meaningful to the contract. The following apply:

- a. The software to which the standard is applied may or may not be designated as Computer Software Configuration Items (CSCIs). While this standard is written in terms of CSCIs, it may be applied to any software, with the term "CSCI" interpreted appropriately.
- b. If not otherwise specified in the contract, this standard shall be understood to apply to the deliverable application software only. This statement is provided as a default; the contracting agency is encouraged to specify the applicability of the standard in contract-specific terms.

1.2.2 Tailoring. This standard and its Data Item Descriptions (DIDs) are meant to be tailored for each item or type of software to which they are applied. The tailoring process for the standard includes deletion of non-applicable requirements and modification of requirements not meeting contract needs. This tailoring, together with additional requirements not provided by this standard, is specified in the Statement of Work. The tailoring process within the DIDs is limited by law to deletion of non-applicable requirements, but can include formatting choices, such as putting two documents under one cover, that do not increase the DID requirements. This tailoring is specified in the Contract Data Requirements List (CDRL). While tailoring is the responsibility of the contracting agency, suggested tailoring may be provided by the contractor. Tailoring guidance is provided in Appendixes G.

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 agency, the term "subcontractor" refers to any contractor of that agency, and the terms "Government" and "contracting agency" apply to the sponsor or other agency with oversight responsibility, as applicable.

1.2.4 References to the CDRL. When this standard is applied in the absence of a Contract Data Requirements List (CDRL) (for example, on Government in-house projects, subcontracts, or contractual vehicles such as task orders without CDRLs),

the term CDRL shall be interpreted to mean the list or itemization of deliverables required from the contractor.

- 1.2.5 System development.** Paragraphs regarding system-level activities shall be interpreted as follows: If the software covered by this standard is part of a larger system for which system-level activities are specified elsewhere (such as MIL-STD-499), the term "participate in" applies as stated. If the software (possibly with its computers) is considered to constitute a system and system-level activities are not specified elsewhere, the term "participate in" shall be interpreted as "be responsible for."
- 1.2.6 Meaning of "develop" and "define".** Throughout this standard, the terms "develop" and "define" shall be interpreted to include creating new items, incorporating existing items, updating or refining existing items, or a combination of these approaches.
- 1.2.7 Meaning of "record".** Throughout this standard, the term "to record" shall be interpreted to mean "to set down in a manner that can be retrieved and viewed." The result may take many forms, including, but not limited to, hand-written notes, hardcopy or electronic documents, and data recorded in computer-aided software engineering (CASE) and project management tools.

2. REFERENCED DOCUMENTS

2.1 Government documents.

2.1.1 Specifications, standards, and handbooks.

| <u>Document Name:</u> | <u>Supply Source:</u> |
|---|-----------------------|
| a. G-S Project Management Manual | G-SII |
| b. USCG Information System Technical Architecture (COMDTINST TBD) | G-SIA |
| c. USCG Software Development and Documentation Standards (COMDTINST 5234.4) | G-SCC-2 |
| d. USCG Software Engineering Guidelines (COMDTINST TBD) | G-SCC-2 |
| e. USCG Data Element Naming Standards, (COMDTINST 5230.42A) | G-SIA |
| f. GSA's - A Guide for Acquiring Federal Information Processing Support Services | G-SIA |
| g. Customer Acquisition Handbook for Non-Major Systems | G-ACS |
| h. Guide for Requirements Analysis and Analysis of Alternative, U.S. General Services Administration, January 1990 (Document Ordering Number: (KMP-90-2P) | G-SIA |
| g. USCG Automated Information Systems (AIS) Security Manual, (COMDTINST M5500.13A) | G-SII |

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

2.2 Non-Government 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. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

3. DEFINITIONS

- 3.1 **Acceptance**. The act of an authorized representative of the Government by which the Government assumes ownership of supplies as partial or complete performance of the contract. (Excerpt from FAR 46.101)
- 3.2 **Application software**. Software designed to fulfill specific needs of a user; for example, software for navigation, payroll, or process control. Contrast with: support software; system software.
- 3.3 **Approval**. Written notification that a contractor's plans, design, or other aspects of the project appear to be sound and can be used as the basis for further work. Such approval in no way shifts responsibility from the contractor to meet contractual requirements.
- 3.4 **Behavioral design**. The external appearance and behavior selected to respond to an item's requirements, ignoring the internal architecture of the item.
- 3.5 **Commercial off-the-shelf (COTS) item**. A item produced and placed in stock by a distributor before receiving orders or contracts for its sale. (Excerpt from FAR 46.101)
- 3.6 **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.
- 3.7 **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.
- 3.8 **Computer resources**. The totality of computer hardware, software, personnel, documentation, supplies, and services applied to a given effort.
- 3.9 **Computer software (or software)**. A combination of associated computer instructions and computer data definitions required to enable computer hardware to perform computational or control functions.

- 3.10 **Computer Software Configuration Item (CSCI)**. An aggregation of software that satisfies an end use function and is designated by the Government for separate configuration management. CSCIs are selected based on tradeoffs among software function, size, host or target computers, developer, support concept, plans for reuse, criticality, interface considerations, need to be separately documented and controlled, and other factors.
- 3.11 **Configuration Item**. An aggregation of hardware, software, or both that satisfies an end use function and is designated by the Government for separate configuration management.
- 3.12 **Contracting agency**. As used in this standard, the contracting office as defined in FAR Subpart 2.1, or its designated representative.
- 3.13 **Data bank**. A collection of data, often a flat file, usually containing data for only one application and not supported by a Database Management System (DBMS). Contrast with database.
- 3.14 **Database**. A collection of data, often containing data for more than one application and usually supported by a Database Management System (DBMS). Contrast with data bank.
- 3.15 **Document**. A data medium and the data recorded on it, that generally has permanence and that can be read by humans or machines.
- 3.16 **Evaluation**. The process of determining whether an item or activity meets specified criteria.
- 3.17 **Firmware**. The combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device. The software cannot be readily modified under program control.
- 3.18 **Hardware Configuration Item (HWCI)**. An aggregation of hardware that satisfies an end use function and is designated by the Government for separate configuration management.
- 3.19 **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.

- 3.20 **Joint review.** A review involving representatives of both the Government and the contractor.
- 3.21 **Life cycle model.** A framework containing the processes, activities, and tasks involved in the development, operation, and support of a system, spanning the life of the system from the definition of its requirements to the termination of its use.
- 3.22 **Non-developmental item.** Any item that: 1) Is available in the commercial marketplace; 2) Was previously developed and is in use by a Federal, State, or local agency of the U.S. or a foreign government with which the U.S. has a mutual defense cooperation agreement; 3) Fits criterion 1 or 2 and requires only minor modification to meet the needs of the procuring agency; or 4) Is currently being produced and will meet criterion 1 or 2 when available or in use. (Excerpt from DODI 5000.2)
- 3.23 **Non-developmental software (NDS).** A non-developmental item that is software.
- 3.24 **Off-the-shelf item.** An item produced and placed in stock by a contractor or stocked by a distributor, before receiving orders or contracts for its sale. The item may be commercial or produced to military or Federal specifications or description. (Excerpt from FAR 46.101)
- 3.25 **Process.** An organized set of activities performed for a given purpose; for example, the software development process.
- 3.26 **Reusable software.** 1) 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.
2) Software developed with the intent of being usable in more than one application.
- 3.27 **Review.** A process or meeting during which a work product or set of work products is presented to project personnel, managers, users, customers, or other interested parties for comment or approval.
- 3.28 **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, rationale, and constraints; design documentation

and data; schedule and status information; and test requirements, test cases, test procedures, and test results.

- 3.29 **Software development library (SDL)**. A controlled collection of software, documentation, other intermediate and final software development products, and associated tools and procedures used to facilitate the orderly development and subsequent support of software. The SDL includes the items under contractor configuration control as part of its contents. A software development library provides storage of and controlled access to software development products in human-readable form, machine-readable form, or both. The library may also contain management data pertinent to the software development project.
- 3.30 **Software development product**. Software and associated data created, modified, or incorporated to satisfy a software development contract. Examples include plans, requirements, design, code, code listings, test information, and manuals.
- 3.31 **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 computer-aided software engineering (CASE) tools, compilers, assemblers, linkers, loaders, operating system, debuggers, simulators, emulators, test tools, documentation tools, and database management systems.
- 3.32 **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.
- 3.33 **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, test case generators, path analyzers, etc. and may also include those tools used in the software engineering environment.
- 3.34 **Software unit**. A logical element in the design of a CSCI; for example, a class, object, module, function, routine, or database. When implemented in code, a logical grouping of computer instructions, data definitions, and/or data.
- 3.35 **Support software**. Software that aids in the development or maintenance of other software, for example compilers, loaders, and

other utilities. Contrast with application software; system software.

- 3.36 **System software**. Software designed to facilitate the operation and maintenance of a computer system; for example operating systems, diagnostic software, and other utilities. Contrast with: application software; support software.
- 3.37 **Validation**. The process of evaluating software to determine compliance with specified requirements.
- 3.38 **Vendor software development kit**. Information and possibly tools provided by a vendor of commercial off-the-shelf (COTS) software to enable purchasers of the software to incorporate, interface other software with, and sometimes, modify the COTS software.
- 3.39 **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.
- 3.40 **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.
- 3.41 **Definitions of acronyms used in this standard**. See Appendix A.

4. GENERAL REQUIREMENTS

4.1 Software development process. The contractor shall establish a software development process consistent with contract requirements. If the contract establishes a life cycle model for the software, the contractor shall map the activities required by this standard onto that portion of the life cycle model covered by the contract. If the contract does not establish a life cycle model for the software, the contractor shall develop a proposed life cycle model consistent with contract requirements, perform the mapping described above, and coordinate the results with the contracting agency. The software development process shall include the following major activities, which may overlap, may begin and end in any order, may be applied iteratively, and may be applied differently to different elements of software. Appendix G provides guidance on applying CG-SDDS to various life cycle models.

- a. Project planning (section 5.1)
- b. Establishing a software development environment (5.2)
- c. System requirements analysis (5.3)
- d. System design (5.4)
- e. Software requirements analysis (5.5)
- f. Software architectural design (5.6)
- g. Software detailed design (5.7)
- h. Coding and unit testing (5.8)
- i. Unit integration and testing (5.9)
- j. CSCI testing (5.10)
- k. CSCI integration and testing (5.11)
- l. System testing (5.12)
- m. Final acceptance testing (5.13)
- n. Preparation for software use and support (5.14)
- o. Preparation for software delivery (5.15)
- p. Integral processes:
 - 1) Software product evaluation (5.16)
 - 2) Software configuration management (5.17)
 - 3) Corrective action and process improvement (5.18)
 - 4) Joint reviews (5.19)
 - 5) Other management activities (5.20)
 - 6) Software quality assurance (5.21)

4.2 General requirements for software engineering.

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 reviews and audits required by the contract.

4.2.2 Reusable software. The contractor shall evaluate and incorporate reusable software in accordance with the following requirements.

4.2.2.1 Evaluating reusable software. The contractor shall evaluate reusable software for use in product solutions and in the software engineering and test environments. Criteria for suitability shall be the software's ability to meet user needs and to be cost-effective over the life of the system, including, as applicable:

- a. Ability to perform required capabilities
- b. Reliability/maturity, as evidenced by established track record
- c. Security and privacy
- d. Safety
- e. Testability, including ability to allocate requirements to the software
- f. Interoperability with other system and system-external elements
- g. Fielding issues, including:
 - 1) Restrictions on copying/distributing the software or documentation
 - 2) License or other fees applicable to each copy
- h. Maintainability, including:
 - 1) Likelihood the software will need to be changed
 - 2) Feasibility of accomplishing that change
 - 3) Available documentation and source code
 - 4) Likelihood that the current version will continue to be supported by the supplier
 - 5) Impact on the system if that version is not supported
 - 6) The Government's data rights to the software
 - 7) Warranties available
- i. Short and long-term cost impacts of using the software
- j. Technical, cost, and schedule risks and tradeoffs in using the software

4.2.2.2 Incorporating reusable software. The contractor shall incorporate reusable software that, based on the contractor's analysis, will meet user needs and be cost-effective over the life of the system. The contractor shall communicate to the contracting agency, via written plans, design documents, or other means, plans for incorporating reusable software. Appendix I provides guidelines for interpreting this standard as applied to reusable software.

Incorporation of reusable software shall comply with the data rights requirements in the contract.

4.2.3 Safety analysis. Safety shall be a principal concern in the design and development of the system and its associated software development products. The contractor shall establish a software safety program and shall integrate it with the system safety program if one exists. The contractor shall perform the analyses and tests necessary to ensure that the requirements, design, code, and operating procedures minimize or eliminate the potential for hazardous conditions during the operational mission. Any potentially hazardous conditions or operating procedures shall be identified in the software requirements, software design, software product specifications, version descriptions, and appropriate user/operator manuals.

4.2.4 Processing resource and reserve capacity. The contractor shall analyze the processing resource and reserve requirements, such as timing, memory utilization, and 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 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 for each CSCI.

4.2.5 Recording rationale. The contractor shall record the rationale for key decisions made in specifying, designing, implementing, and testing the software. This rationale shall include trade-offs considered, analysis methods, and criteria used to make the decisions. The rationale shall be recorded in documents, code comments, or other media that will transition to the support agency.

4.3 General requirements for software testing.

4.3.1 Independence in testing activities. The persons who plan and perform CSCI, CSCI integration, and system testing shall not be the persons who developed the software being tested. This does not preclude the persons who developed the software from participating in these activities (for example, identifying supplemental tests that rely on knowledge of CSCI internal design).

- 4.3.2 Testing on target computer system.** The contractor's CSCI, CSCI integration, and system testing shall include testing on the target computer system or an equivalent system approved by the contracting agency.
- 4.3.3 Stress testing.** The contractor's CSCI, CSCI integration, and system testing shall include stressing the software at the limits of its specified requirements.

5. DETAILED REQUIREMENTS

The requirements in this section may be carried out in any order consistent with contract requirements. Figure 1 shows a framework for applying the activities across multiple builds. Appendix G explains the concept of builds. A contract involving a single build is one example of this framework. If the software is developed in multiple builds, some activities may be performed in every build, others only in selected builds, and activities and products may not be complete until several or all builds are accomplished. Regardless of the number of builds, many of these activities may be ongoing at one time, and activities specified in early sections may depend on input from activities in later sections. There is no intent to specify a sequence. Appendix G provides guidelines for tailoring and scheduling these requirements.

| ACTIVITY | INCREMENTAL /BUILDS | INCREMENTAL /BUILDS | | | |
|----------|---|---------------------|------------|------------|------------|
| | | BUILD 1 | BUILD 2 | BUILD 3 | BUILD 4 |
| 5.1 | Project Planning | | | | |
| 5.2 | Establishing a SW development environment | | | | |
| 5.3 | System requirements analysis | | | | |
| 5.4 | System design | | | | |
| 5.5 | Software requirement analysis | | | | |
| 5.6 | Software architectural design | | | | |
| 5.7 | Software detailed design | | | | |
| 5.8 | Coding and unit testing | | | | |
| 5.9 | Unit integration and testing | | | | |
| 5.10 | CSCI testing | | | | |
| 5.11 | CSCI integration and testing | | | | |
| 5.12 | System testing | | | | |
| 5.13 | Final Acceptance Testing | | | | |
| 5.14 | Preparation for software use and support | | | | |
| 5.15 | Preparation for software delivery | | | | |
| | Integral processes: | | | | |
| 5.16 | Software product evaluations | | | | |
| 5.17 | Software configuration management | | | | |
| 5.18 | Corrective action & process improvement | | | | |
| 5.19 | Joint reviews | | | | |
| 5.20 | Other management activities | | | | |
| 5.21 | Software Quality Assurance | | | | |

FIGURE 1. Framework for applying CG-SDDS across multiple builds.

FIGURE 1. Framework for applying CG-SDDS across multiple builds.

5.1 Project planning. The contractor shall perform the following planning activities during the course of the contract. Following contracting agency approval of any of these plans, the contractor shall conduct the relevant activities in accordance with the plan and, with the exception of contractor-internal scheduling information, updates shall be subject to contracting agency approval.

Note: If a system is developed in multiple builds, planning for each build should be interpreted to mean planning that build in detail and planning future builds covered under the contract to a level of detail compatible with the information available.

5.1.1 Software development planning. The contractor shall develop and record plans for conducting the activities required by this standard. This planning shall include all applicable items in the Coast Guard Software Plan (CG-SP) DID (see 6.3).

Note 1: If the CDRL specifies delivery of the data generated by this or any other paragraph, the contractor is required to format, assemble, mark, copy, and distribute the deliverable in accordance with the CDRL. This task is recognized to be separate from the task of generating the required information and to require additional time and effort on the part of the contractor.

Note 2: The wording here and throughout CG-SDDS is designed to: *Emphasize that the development and recording of planning and engineering information is an intrinsic part of the software development process, to be performed regardless of whether a deliverable is required; 2) Use the DID as a checklist of items to be covered in the planning or engineering activity; and 3) Permit representations other than traditional documents for recording the information (e.g., computer-aided software engineering (CASE) tools).*

5.1.2 CSCI and CSCI integration test planning. The contractor shall develop and record plans for conducting the CSCI and CSCI integration testing required by this standard. This planning shall include all applicable items in the Coast Guard Software Test Document (CG-STD) DID (see 6.3).

5.1.3 System test planning. The contractor shall participate in developing and recording plans for conducting the system-level testing required by this standard. For software systems, this planning shall include all applicable items in the Coast Guard Software Test Document (CG-STD) DID (see 6.3).

- 5.1.4 Software support planning.** The contractor shall identify all resources used or generated during software development that will be needed by the support agency. The contractor shall develop and record plans for transitioning the deliverable software and associated elements of the software engineering and test environments to the support agency. This planning shall include all applicable items in the Coast Guard Software Plan (CG-SP) DID (see 6.3).
- 5.1.5 Software installation planning.** The contractor shall develop and record plans for performing software installation and training at the user sites specified in the contract. This planning shall include all applicable items in the Coast Guard Software Plan (CG-SP) DID (see 6.3).
- 5.2 Establishing a software development environment.** The contractor shall perform the following activities during the course of the contract.
- Note:** If a system or CSCI is developed in multiple builds, establishing the environment in each build should be interpreted to mean establishing the environment needed to complete that build.
- 5.2.1 Software engineering environment.** The contractor shall establish, control, and maintain a software engineering environment to perform the software engineering effort. The software engineering environment shall comply with the security requirements of the contract.
- 5.2.2 Software test environment.** The contractor shall establish, control, and maintain a software test environment to perform the software testing effort. The software test environment shall comply with the security requirements of the contract. Following installation, each item of the environment shall be tested to demonstrate that the item performs its intended function.
- 5.2.3 Software development library.** The contractor shall establish, control, and maintain a software development library (SDL) to provide storage of, and controlled access to, intermediate and final products of the software development process. The SDL may be an integral part of the software engineering and test environments. The contractor shall maintain the SDL for the duration of the contract.

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

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

5.2.5 Design and coding standards.

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

5.2.6 Non-deliverable software. Non-deliverable software may be employed in the development of deliverable software as long as the operation and support of the deliverable software after its delivery to the Government are independent of such software.

5.3 System requirements analysis. The contractor shall perform the following system requirements analysis activities during the course of the contract (see 1.2.5).

Note: If a system is developed in multiple builds, its requirements may not be fully defined until most or all builds are complete. The contractor's planning should identify the subset of system requirements to be defined in each build and the subset to be implemented in each build. System requirements analysis for a given build should be interpreted to mean defining the system requirements so identified for that build.

- 5.3.1 Analysis of user input.** The contractor shall analyze user input provided by the contracting agency. This input may take the form of need statements, surveys, problem/change reports, feedback on prototypes, interviews, or other user input or feedback.
- 5.3.2 Operational concept.** The contractor shall participate in defining and recording the operational concept for the system. The analysis shall include all applicable items in the Coast Guard Software Requirements Document (CG-SRD) DID (see 6.3). Participation may take the form of interviewing users, performing domain analysis, modeling the user's environment and needs, describing software capabilities, developing software prototypes, evaluating reusable software components, performing trade-off studies, and other activities to explore feasibility, user preferences, cost effectiveness, risk reduction, and other issues.
- 5.3.3 System requirements.** The contractor shall participate in defining and recording the engineering, interface, and qualification requirements for the system. The analysis shall take into account the anticipated growth envelope of the system and shall include all applicable items in the Coast Guard Software Requirements Document (CG-SRD) DID (see 6.3). Participation may take the form of specifying system requirements involving software capabilities, developing software prototypes, performing trade-off studies, evaluating reusable software components, and other activities to explore feasibility, user preferences, cost effectiveness, risk reduction, and other issues.

Note 1: When approved by the contracting agency, the Coast Guard Software Requirements Document (CG-SRD) establishes the Functional Baseline for a system or segment. If the CG-SRD is already under Government Configuration Management (CM), updates shall be proposed and implemented in accordance with contractually established CM procedures.

- 5.4 System design.** The contractor shall perform the following system design activities during the course of the contract (see 1.2.5).

Note: If a system is developed in multiple builds, its design may not be fully defined until most or all builds are complete. The contractor's planning should identify the portion of the system design to be defined in each build. System design for a given build should be interpreted to mean defining the portion of the system design identified for that build.

- 5.4.1 System behavioral design.** The contractor shall participate in defining and recording those aspects of the system design that transcend the system architecture and describe overall system appearance and behavior in response to system requirements. The analysis shall include all applicable items in the behavioral design portion of the Coast Guard Software Design Document (CG-SDD) DID (see 6.3). This activity shall take into account candidate reusable software components and the anticipated growth envelope of the system, and may include developing software prototypes, performing trade-off studies, and other activities to explore feasibility, user preferences, cost effectiveness, risk reduction, and other issues.
- 5.4.2 System architectural design.** The contractor shall participate in defining and recording the allocation of system requirements to Hardware Configuration Items (HWCIs), Computer Software Configuration Items (CSCIs), and manual operations. The analysis shall include all applicable items except the behavioral design in the Coast Guard Software Design Document (CG-SDD) DID (see 6.3). This activity shall take into account candidate reusable software components and the anticipated growth envelope of the system, and may include exploration of CSCI architecture, development of prototypes, trade-off studies, and other activities to explore feasibility, user preferences, cost effectiveness, risk reduction, and other issues.

5.5 Software requirements analysis. The contractor shall perform the following software requirements analysis activities during the course of the contract:

Note: If a CSCI is developed in multiple builds, its requirements may not be fully defined until most or all builds are complete. The contractor's planning should identify the subset of each CSCI's requirements to be defined during each build and the subset to be implemented during each build. Software requirements analysis for a given build should be interpreted to mean defining the CSCI requirements so identified for that build.

5.5.1 CSCI engineering requirements. The contractor shall define and record a complete set of engineering requirements for each CSCI, the qualification requirements for validating compliance with each engineering requirement, and the traceability between these requirements and the system requirements allocated to the CSCI. The analysis shall include all applicable items in the Coast Guard Software Requirements Document (CG-SRD) DID (see 6.3).

Note: When approved by the contracting agency, the Coast Guard Software Requirements Document (CG-SRD) forms part of the Allocated Baseline for the CSCIs. If an CG-SRD is already under Government Configuration Management (CM), updates shall be proposed and implemented in accordance with contractually established CM procedures.

5.5.2 CSCI interface requirements. The contractor shall define and record a complete set of interface requirements for each interface external to each CSCI, and the qualification requirements for validating compliance with each interface requirement. The analysis shall include all applicable items in the Coast Guard Software Requirements Document (CG-SRD) DID (see 6.3).

Note: When approved by the contracting agency, the Coast Guard Software Requirements Document (CG-SRD) forms part of the Allocated Baseline for the CSCIs. If an CG-SRD is already under Government Configuration Management (CM), updates shall be proposed and implemented in accordance with contractually established CM procedures.

5.6 Software architectural design. The contractor shall perform the following software architectural design activities during the course of the contract.

Note: If a CSCI is developed in multiple builds, its behavioral and architectural design may not be fully defined until most or all builds are complete. Behavioral and architectural design in each build should be interpreted to mean those necessary to meet the CSCI requirements to be implemented during that build.

5.6.1 CSCI behavioral design. The contractor shall define and record those aspects of each CSCI's design that transcend the CSCI architecture and describe overall CSCI appearance and behavior in response to CSCI requirements. The analysis shall include all applicable items in the behavioral design portion of the Coast Guard Software Design Document (CG-SDD) DID (see 6.3).

5.6.2 CSCI architectural design. The contractor shall develop and record the architectural design of each CSCI. The architectural design shall partition each CSCI into software units in accordance with the development methods described in the software development plan. The contractor shall allocate the CSCI requirements to the software units, establish design requirements, as applicable, for further design of the units, and record the traceability between CSCI requirements and the units. The analysis shall include all applicable items in the architectural design portion of the Coast Guard Software Design Document (CG-SDD) DID (see 6.3).

Note: DOD-STD-2167A's computer software components and computer software units are one implementation of this requirement. Contractors who wish to continue using this method may do so.

5.6.3 Database logical design. The contractor shall define and record the logical design of each database included in, or used by, each CSCI. The analysis shall include all applicable items in the logical design portions of the Coast Guard Software Design Document (CG-SDD) DID (see 6.3).

5.7 Software detailed design. The contractor shall perform the following detailed design activities during the course of the contract.

Note: If a CSCI is developed in multiple builds, its detailed design may not be completed until all builds are complete. Detailed design in each build should be interpreted to mean that necessary to meet the CSCI requirements to be implemented during that build.

- 5.7.1 CSCI detailed design.** The contractor shall develop and record the detailed design of each software unit. The analysis shall include all applicable items in the detailed design portion of the Coast Guard Software Design Document (CG-SDD) DID (see 6.3).
- 5.7.2 CSCI interface design.** The contractor shall develop and record the design of the interfaces external to each CSCI. The analysis shall include all applicable items in the Coast Guard Software Design Document (CG-SDD) DID (see 6.3).
- 5.7.3 Database physical design.** The contractor shall develop the physical design of each database included in, or used by, each CSCI. The analysis shall include all applicable items in the physical design portions of the Coast Guard Software Design Document (CG-SDD) DID (see 6.3).
- 5.8 Coding and unit testing.** The contractor shall perform the following coding and unit testing activities during the course of the contract.

Note 1: If a CSCI is developed in multiple builds, coding and unit testing of that CSCI will not be complete until all builds are complete. Coding and unit testing in each build should be interpreted to include those units that need to be developed and tested to meet the CSCI requirements to be implemented during that build.

Note 2: "Coding" as used here means the generation of computer-readable instructions and data definitions in a form, or translatable into a form, that can be acted on by a computer. Examples are statements in a traditional programming language, non-procedural commands in a fourth-generation language, rules in an expert system, and data definitions.

- 5.8.1 Coding software units.**
- a. The contractor shall develop and record code corresponding to each software unit.
 - b. If no high order language (HOL) is required by the contract, the contractor shall obtain contracting agency approval to use a particular language.
- 5.8.2 Populating databases.** For databases to be populated as part of the software development effort, the contractor shall

define the data to be entered, obtain or generate the data, validate the data, and enter it into each database.

- 5.8.3 Preparing for unit testing.** The contractor shall establish test requirements, test responsibilities, test cases (in terms of inputs, expected results, and evaluation criteria), test procedures, and schedules for testing the code corresponding to each software unit. The contractor shall prepare the test data needed to perform the test cases. The contractor's unit 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 software development files for the software units.
- 5.8.4 Performing unit testing.** The contractor shall test the code corresponding to each software unit, ensuring that the algorithms and logic employed by the code are correct and that the code satisfies its specified requirements.
- 5.8.5 Revision and retesting.** The contractor shall make all necessary revisions to the design and code, perform all necessary retesting, and update the software development files of all software units that undergo design or coding changes based on unit tests.

5.8.6 Recording unit test results. The contractor shall record the results of all unit testing in the corresponding software development files (SDFs).

5.9 Unit integration and testing. The contractor shall perform the following unit integration and testing activities during the course of the contract.

Note: If a CSCI is developed in multiple builds, unit integration and testing of that CSCI will not be completed until all builds are complete. Unit integration and testing in each build should be interpreted to mean integrating the code for the units developed in the current build with the code developed in previous builds and testing the results.

5.9.1 Preparing for unit integration and testing. The contractor shall establish test requirements, test responsibilities, test cases (in terms of inputs, expected results, and evaluation criteria), test procedures, and schedules for conducting unit integration and testing. The contractor shall prepare the test data needed to perform the test cases. The 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 appropriate software development files. The analysis shall include all applicable items in the detailed design portion of the Coast Guard Software Test Plan (CG-STP) DID (see 6.3).

5.9.2 Performing unit integration and testing. The contractor shall progressively integrate the code corresponding to tested software units with code for previously integrated units, and test the resulting integrated code until the code for all units in each CSCI is integrated and the resulting CSCI is ready for CSCI testing. This testing shall ensure that the code for the units works together as intended without degrading the required capabilities or performance of the code for the individual units.

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

5.9.4 Recording unit integration and test results. The contractor shall record the results of all unit integration and testing in appropriate software development files (SDFs).

5.10 CSCI testing. The contractor shall perform the following CSCI testing activities during the course of the contract.

Note: If a CSCI is developed in multiple builds, its CSCI testing will not be completed until the final build for that CSCI, or possibly until later builds involving items with which the CSCI is required to interface. CSCI testing in each build should be interpreted to mean planning and performing tests of the current build of each CSCI to ensure compliance with the CSCI requirements established for that build.

- 5.10.1 Preparing CSCI test cases.** The contractor shall develop and record the test cases to be used for testing each CSCI. The contractor shall record the traceability from CSCI requirements to the test cases and shall generate the test data needed to perform the test cases. The analysis shall include all applicable items except test procedures in the Coast Guard Software Test Document (CG-STD) DID (see 6.3).
- 5.10.2 Preparing CSCI test procedures.** The contractor shall develop and record procedures for CSCI testing. The analysis shall include all applicable procedures in the Coast Guard Software Test Document (CG-STD) DID (see 6.3).
- 5.10.3 Dry run of CSCI test procedures.** Prior to Government-witnessed testing of each CSCI, the contractor shall dry run the CSCI test procedures to ensure that the procedures are complete and accurate and that the software is ready for testing. The contractor shall record the results of this activity in the corresponding CSCI software development files (SDFs) and shall update the CSCI test cases and procedures as appropriate.
- 5.10.4 Performing CSCI testing.** The contractor shall test each CSCI in accordance with the CSCI test cases and procedures. The contracting agency shall be notified at least 30 days in advance of the testing.
- 5.10.5 Revision and retesting.** The contractor shall make necessary revisions to the design and code, conduct all necessary retesting, and update the software development files (SDFs) of all software units and CSCIs that undergo changes based on CSCI testing.
- 5.10.6 Analyzing CSCI test results.** The contractor shall analyze and record the results of CSCI testing. The analysis shall include

all applicable items in the Coast Guard Software Test Document (CG-STD) DID (see 6.3).

- 5.11 CSCI integration and testing.** The contractor shall perform the following CSCI integration and testing activities during the course of the contract.

Note: If a system or CSCI is developed in multiple builds, CSCI integration and testing may not be complete until builds are complete. CSCI integration and testing in each build should be interpreted to mean integrating those CSCIs or interim builds of CSCIs with other CSCIs, HWCIs, or interim builds of either as necessary to meet the system requirements to be implemented in that build.

- 5.11.1 Preparing CSCI integration and test cases.** The contractor shall develop and record the test cases to be used for each CSCI integration test. The contractor shall record the traceability from CSCI requirements to the test cases and shall generate the test data needed to perform the test cases. The analysis shall include all applicable items except test procedures in the Coast Guard Software Test Document (CG-STD) DID (see 6.3).
- 5.11.2 Preparing CSCI integration and test procedures.** The contractor shall develop and record procedures for CSCI integration testing. The analysis shall include all applicable procedures in the Coast Guard Software Test Document (CG-STD) DID (see 6.3).
- 5.11.3 Dry run of CSCI integration and test procedures.** Prior to Government-witnessed CSCI integration testing, the contractor shall dry run the CSCI integration and test procedures to ensure that the procedures are complete and accurate and that the software is ready for testing. The contractor shall record the results of this activity in the corresponding CSCI software development files (SDFs) and shall update the CSCI integration and test cases and procedures as appropriate.
- 5.11.4 Performing CSCI integration and testing.** The contractor shall progressively integrate CSCIs with other CSCIs and HWCIs, as applicable, and test the resulting groups in accordance with the CSCI integration test cases and procedures, until all CSCIs for which the contractor is responsible have been integrated and tested. This testing shall ensure that the CSCIs and HWCIs work together as intended without degrading the required capabilities or performance

of the individual CSCIs. The contracting agency shall be notified at least 30 days in advance of the testing.

- 5.11.5 **Revision and retesting.** The contractor shall make necessary revisions to the design and code, conduct all necessary retesting, and update the software development files (SDFs) of all CSCIs that undergo changes based on CSCI integration and testing.
- 5.11.6 **Analyzing CSCI integration and test results.** The contractor shall analyze and record the results of CSCI integration and testing. The analysis shall include all applicable items in the Coast Guard Software Test Document (CG-STD) DID (see 6.3).
- 5.12 **System testing.** The contractor shall perform the following system testing activities during the course of the contract (see 1.2.5).

Note: If a system is developed in multiple builds, testing the completed system will not occur until the final build of that system. System testing in each build should be interpreted to mean planning and performing tests of the current build of the system to ensure compliance with the system requirements established for that build.

- 5.12.1 **Preparing system test cases.** The contractor shall participate in developing and recording the test cases to be used for system testing. The contractor shall participate in recording the traceability from system requirements to the test cases and in generating the test data needed to perform the test cases. For software systems, the analysis shall include all applicable items except test procedures in the Coast Guard Software Test Document (CG-STD) DID (see 6.3).
- 5.12.2 **Preparing system test procedures.** The contractor shall participate in developing and recording procedures for system testing. For software systems, the analysis shall include all applicable procedures in the Software Test Description (CG-STD) DID (see 6.3).
- 5.12.3 **Dry run of system test procedures.** Prior to Government-witnessed system testing, the contractor shall participate in dry running the system test procedures to ensure that the procedures are complete and accurate and that the system is ready for testing. The contractor shall record the software-related results of this activity in the corresponding CSCI software development files (SDFs)

and shall participate in updating the system test cases and procedures as appropriate.

- 5.12.4 **Performing system testing.** The contractor shall participate in system testing activities in accordance with the system test cases and procedures. The contracting agency shall be notified at least 30 days in advance of the testing.
- 5.12.5 **Revisions and retesting.** The contractor shall make necessary revisions to the design and code, conduct all necessary retesting, and update the software development files (SDFs) of all software units and CSCIs that undergo changes based on system testing.
- 5.12.6 **Analyzing system test results.** The contractor shall participate in analyzing system test results and recording the results. For software systems, the analysis shall include all applicable items in the Coast Guard Software Test Document (CG-STD) DID (see 6.3).
- 5.13 **Final acceptance testing.** The development contractor shall prepare the system for final acceptance testing. The purpose of Final acceptance testing is to determine if the product complies with the system requirements and is acceptable to the acquirer, as is. This testing is not performed by the development contractor. It will be done by an agency or team that is separate from and independent of, the contractor development team.
 - 5.13.1 **Preparing for final acceptance testing.** The development contractor shall deliver a complete system, as per the contract to the independent tester. The delivery shall include all software necessary to run the system, along with any required manuals or documentation. The development contractor shall also deliver a copy of all previous test plans and procedures, a copy of previous test findings and a report of all corrective actions taken, prior to the start of final acceptance testing.
 - 5.13.2 **Preparing final acceptance test procedures.** The designated independent tester (DIT) shall develop and record procedures for Final Acceptance testing. These procedures shall confirm that all previous testing results continue to meet system and functional requirements, as well as test the entire system as a whole.
 - 5.13.3 **Dry run of final acceptance test procedures.** Prior to the government witnessed final acceptance testing, the DIT shall dry run

the final acceptance test procedures to ensure that the procedures are complete and accurate, and that the system is ready for testing. The DIT shall coordinate with the development contractor to ensure that the results of this activity, are recorded in the corresponding software development files and updates to the final acceptance test cases and procedures are made as appropriate.

- 5.13.4 Performing final acceptance testing.** The DIT shall progressively test each CSCI in the system, as well as the integration of each CSCI to the system in accordance with the Final Acceptance Test procedures. This testing shall ensure that the CSCI's and HWCI's work together as intended without degrading the required capabilities or performances of the individual CSCI's.
- 5.13.5 Revisions and retesting.** It is the responsibility of the DIT to notify the system acquirer of the results of final acceptance testing. The development contractor shall make the required revisions to the design and code, and submit the revised software for all necessary retesting. The development contractor shall annotate and update all effected documentation that undergoes changes based on final acceptance results.
- 5.13.6 Analyzing final acceptance test results.** The DIT, in conjunction with the development contractor shall analyze and record the results of final acceptance testing. The analysis shall include all applicable items in the Coast Guard Software Test Document (CG-STD) DID (see 6.3)
- 5.14 Preparation for software use and support.** The contractor shall perform the following activities during the course of the contract.

Note: The manuals in this section are normally developed in parallel with software development, ready for use and validation in CSCI testing. If a system or software is developed in multiple builds, the contractor's planning should identify what software, if any, is to be fielded to users each build, the extent of fielding (for example, full fielding or fielding to selected evaluators only), and what software, if any, is to be transitioned to the support agency each build. Preparation of manuals and performance of installation and transition activities in each build should be interpreted to include those necessary to support the fielding and transition plans for that build.

- 5.14.1 Software user manuals.** The contractor shall identify and record information needed by hands-on users of the software to

install and execute the software. The information shall include all applicable items in the Coast Guard Software User/Operator Manual (CG-SUOM) DID (see 6.3).

- 5.14.2 Computer center software operator manuals.** The contractor shall identify and record information needed by computer center personnel to install and operate the software. The information shall include all applicable items in the Coast Guard Software User/Operator Manual (CG-SUOM) DID (see 6.3).
- 5.14.3 Software input/output manuals.** The contractor shall identify and record information needed by persons who will submit inputs to and receive reports from the software, but will not operate, or be hands-on users of the software. The information shall include all applicable items in the Coast Guard Software User/Operator Manual (CG-SUOM) DID (see 6.3).
- 5.14.4 Computer system operator manuals.** The contractor shall identify and record information needed to operate any new or non-standard computer systems on which the software will run and for which commercial operator manuals are not available. The information shall include all applicable items in the Coast Guard Software User/Operator Manual (CG-SUOM) DID (see 6.3).
- 5.14.5 Computer instruction set architecture manuals.** The contractor shall identify and record information needed to program any new or non-standard computer systems used as the host or target computer for the software, and for which commercial programmer manuals are not available. The information shall include all applicable items in the Coast Guard Software Support Document (CG-SSD) DID (see 6.3).
- 5.14.6 Installation at user sites.** The contractor shall:
- a. Install and check out the operational software at the user sites specified in the contract.
 - b. Provide training to users as specified in the contract.
 - c. Provide other support to user sites as specified in the contract.
- 5.14.7** Transition to designated support site. The contractor shall:
- a. Install and check out the deliverable software in the support environment designated by the contracting agency.

- b. Demonstrate to the contracting agency that the deliverable software can be regenerated (compiled/linked/loaded into an executable product) and maintained using commercially available, Government-owned, or contractually deliverable support software and hardware that has been designated by the contracting agency.
- c. Provide training to the support agency as specified in the contract.
- d. Provide other support to the support agency as specified in the contract.

5.15 Preparation for software delivery. The contractor shall perform the following activities during the course of the contract.

Note: If a system or CSCI is developed in multiple builds, software delivery may take place in each build, in selected builds, or only in the final build. The contractor's planning should identify what software will be delivered in each build. Preparation for delivery in each build should be interpreted to cover the software designated for delivery in that build.

5.15.1 Preparing executable code for delivery. The contractor shall prepare the executable software for delivery as specified in the contract (see 6.2) or the Coast Guard Software Requirements Document (CG-SRD).

5.15.2 Preparing source code for delivery. The contractor shall prepare the source code for each CSCI for delivery as specified in the contract or the Coast Guard Software Requirements Document (CG-SRD).

5.15.3 Software product specifications. The contractor shall prepare the information needed to describe the deliverable software product. This information shall include all applicable items in the Coast Guard Software Support Document (CG-SSD) DID (see 6.3).

Note: When approved by the contracting agency, the Coast Guard Software Support Document (CG-SSD) can establish the Product Baseline for the CSCIs. If a CG-SSD is already under Government Configuration Management (CM), updates shall be proposed and

implemented in accordance with contractually established CM procedures.

- 5.15.4 **Version descriptions.** The contractor shall identify and record the exact version of software to be delivered. The information shall include all applicable items in the Coast Guard Software Support Document (CG-SSD) DID (see 6.3).
- 5.15.5 **Functional Configuration Audits.** The contractor shall support one or more Functional Configuration Audits (FCAs). These audits shall demonstrate the readiness of the software for Government acceptance testing, as applicable.
- 5.15.6 **Physical Configuration Audits.** The contractor shall support one or more Physical Configuration Audits (PCAs).
- 5.16 **Software product evaluations.** The contractor shall perform the following software product evaluation activities during the course of the contract.

Note: If a system or CSCI is developed in multiple builds, the products of each build should be evaluated in light of the stated objectives of that build. A product that meets those objectives can be considered complete and satisfactory even though it is missing information designated for development in later builds.

- 5.16.1 **In-process software product evaluations.** The contractor shall perform in-process evaluations of the software development products generated in carrying out the requirements of this standard. The products to be evaluated, criteria to be used, and definitions for those criteria are given in Appendix D. The contractor may propose additional or alternative criteria and definitions. These criteria and definitions are subject to contracting agency approval.
- 5.16.2 **Final software product 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.
- 5.16.3 **Independence in product evaluation activities.** The persons who evaluate a software development product shall not be the persons who developed the product. This does not preclude the persons who developed the product from participating in the evaluation (for example, as participants in a walk-through of the product).

5.16.4 Software product 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.

5.17 Software configuration management. The contractor shall perform the following software configuration management activities during the course of the contract.

Note 1: The requirements in this section are concerned with: 1) contractor-internal control of intermediate and final software development products, and 2) basic requirements for interfacing with Government Configuration Management.

Note 2: If a system or CSCI is developed in multiple builds, the products of each build may be refinements of, or additions to, previous versions of those products. Incrementally developed specifications may be baselined by the Government in increments, or baselining may be deferred until the specifications are complete. Contractor configuration control in each build must take place in the context of the previous products and Government baselines in place during that build.

5.17.1 Contractor control of development products. The contractor shall establish and implement procedures for identifying and controlling the intermediate and final versions of the software development products generated and used in carrying out the requirements of this standard. This requirement includes all products identified in Appendix D except those corresponding to activities tailored out of this standard and the contents of software development files.

5.17.1.1 Contractor-internal configuration identification. The contractor shall perform configuration identification of intermediate and final software development products. The contractor's configuration identification shall comply with any identification scheme specified in the contract and shall accomplish the following. The word "identify" shall be interpreted to mean "assign a unique number, label, or other identifier to."

- a. Identify the documentation that establishes, or will establish, the Functional, Allocated, and Product Baselines.
- b. Identify each CSCI and its corresponding software units.
- c. Identify the documentation and the computer software media (such as computer disks or files) containing code and other software development products that are to be placed under configuration control.
- d. Identify the version, release, change status, and any other identification details of each item under configuration control.
- e. Identify the specific version of software contained on a deliverable medium, including all changes incorporated since its previous release.

5.17.1.2 Contractor-internal configuration control. The contractor shall perform configuration control of intermediate and final software development products. The contractor's configuration control shall accomplish the following:

- a. Establish and implement procedures designating:
 - 1) The levels of control each software development product must pass through (for example, author control; released for software product evaluation; submitted to software development library; ...)
 - 2) The persons or groups with authority to authorize changes and to make changes at each level (for example, the programmer/analyst, the software lead, the project manager)
 - 3) The steps to be followed to request authorization for change, process such requests, track changes, and maintain past versions.
- b. Maintain or have the ability to recreate copies of delivered software development products.
- c. Provide the contracting agency access to software development products under configuration control.
- d. Control the preparation and dissemination of changes to software development products that have been placed under

configuration control so that current copies reflect approved changes.

- 5.17.2 Interface with Government Configuration Management.** The contractor shall interface with the contracting agency's Configuration Management system as specified in the contract. In the absence of such requirements, the following basic requirements apply.
- 5.17.2.1 Engineering Change Proposals.** If the software development process reveals necessary or beneficial changes to specifications that have been placed under Government Configuration Management, the contractor shall prepare Engineering Change Proposals (ECPs) and Specification Change Notices (SCNs) in accordance with MIL-STD-973, using the regular or short form of the ECP as specified in the contract.
- 5.17.2.2 Configuration status accounting.** The contractor shall generate configuration status reports on all products developed, modified, or held by the contractor comprising the Functional, Allocated and Product Baselines. The status reports shall:
- a. Provide traceability of changes to these baselines.
 - b. Serve as a basis for communicating the status of these baselines and associated software.
 - c. Serve as a vehicle for ensuring that delivered documents describe and represent the associated software.
- 5.17.3 Storage, handling, and delivery of project media.** The contractor shall establish and implement procedures for the storage, handling, and delivery of software development products. The contractor shall maintain master copies of the delivered products.
- 5.18 Corrective action and process improvement.** The contractor shall perform corrective action and process improvement in accordance with the following requirements.
- 5.18.1 Problem/change reports.** The contractor shall prepare a problem/change report to describe each problem detected or change proposed in the processes being used or in development products after they have been released for software product evaluation. The problem/change report shall describe the problem or proposed change, the corrective action or improvement needed, and the actions taken

to date. These reports shall serve as input to the corrective action and process improvement system.

- 5.18.2 Corrective action and process improvement system.** The contractor shall implement a corrective action and process improvement system for improving the processes used in software development and for handling all problems detected and changes recommended in development products after they have been released for software product evaluation. The system shall comply with the following requirements:
- a. Inputs to the system shall consist of problem/change reports, other discrepancy reports, and suggestions for process changes.
 - b. The system shall be closed-loop, ensuring that all detected problems are promptly reported, reported problems and suggested changes are promptly entered into the system, action is initiated on them, resolution is achieved, status is tracked and reported, and records of the problems and changes are maintained for the life of the contract.
 - c. Each problem in software development products shall be classified by category and priority, using the categories and priorities in Appendix C or approved alternatives.
 - d. Analysis shall be performed to detect trends in the problems and suggested changes reported and to identify process improvements that would reverse adverse trends.
 - e. Corrective actions shall be evaluated to: (1) verify that problems have been resolved, adverse trends have been reversed, and changes have been correctly implemented in the appropriate processes and products, and (2) to determine whether additional problems have been introduced.

- 5.19 Joint reviews.** The contractor shall plan and participate in joint (Government/ contractor) reviews in accordance with the following requirements.

Note 1: The requirements in this section are concerned with technical and management level reviews of the software development effort.

Note 2: If a system or CSCI is developed in multiple builds, the types of reviews held will depend on the objectives of each build. When reviews are held on interim products, those products should be reviewed in light of the stated objectives of the current build. Products that meet those objectives can be considered complete and satisfactory even though they are missing information designated for development in later builds.

5.19.1 Joint technical reviews. The contractor shall plan and participate in joint technical reviews at locations and dates proposed by the contractor and approved by the contracting agency. These reviews shall be attended by persons with technical knowledge of the work products to be reviewed. The reviews shall use in-process and final products of software development, rather than materials generated especially for the review. The reviews shall have the following objectives:

- a. Surface and resolve technical issues, using as criteria the software product evaluation criteria in Appendix D.
- b. Surface near-and long-term risks regarding technical, cost, and schedule issues.
- c. Arrive at agreed upon mitigation strategies for those risks, within the authority of those present.
- d. Identify risks and issues to be raised at joint management reviews.
- e. Ensure on-going communication between Government and contractor technical personnel.

5.19.2 Joint management reviews. The contractor shall plan and participate in joint management reviews at locations and dates proposed by the contractor and approved by the contracting agency. These reviews shall be attended by persons with authority to make cost and schedule decisions and shall have the following objectives. Examples of such reviews are identified in Appendix E.

- a. Resolve issues that could not be resolved at joint technical reviews.
- b. Arrive at agreed-upon mitigation strategies for near and long-term risks that could not be resolved at joint technical reviews.
- c. Identify and resolve management-level issues and risks not raised at joint technical reviews.
- d. Obtain Government approvals and commitments needed for timely accomplishment of the project.

5.20 Other management activities. The contractor shall perform the following software development management activities during the course of the contract.

5.20.1 Risk management. The contractor shall:

- a. Identify, analyze, and prioritize the areas of the software development project that involve potential technical, cost, or schedule risks.
- b. Develop strategies for eliminating or reducing those risks.
- c. Incorporate the risks and strategies into software development planning.
- d. Implement the strategies, monitor the risks, and adjust the strategies as needed throughout the software development process.

- 5.20.2 **Security and privacy.** The contractor shall comply with the security and privacy requirements specified in the contract. These requirements may affect the software development effort, the resulting software products, or both.
- 5.20.3 **Subcontractor management.** If subcontractors are used, the contractor shall pass down to the subcontractors all contractual requirements necessary to ensure that software development products are developed in accordance with prime contract requirements.
- 5.20.4 **Interface with software IV&V agents.** The contractor shall interface with the software Independent Verification and Validation (IV&V) agent(s) as specified in the contract.
- 5.20.5 **Interface with associate contractors.** The contractor shall coordinate with associate contractors as required to ensure compatibility of interfacing development products.
- 5.20.6 **Software management indicators.** The contractor shall:

- a. Identify a set of software management indicators to aid in managing the development process and communicating its status to the Government. Candidate indicators are given in Appendix F.
- b. Identify the data to be collected, the methods to be used to interpret and apply the data, and the planned reporting mechanism in the software development plan.
- c. Collect, interpret, apply, and report on those indicators as described in the software development plan.

5.21 Software quality assurance. The contractor shall perform software quality assurance in accordance with the following requirement.

Note: If a system CSCI is developed in multiple builds, the activities and software products of each build should be evaluated in the context of the objectives established for that build. An activity or software product that meets those objectives can be considered satisfactory even though it is missing aspects designated for later builds.

5.21.1 Software quality assurance evaluations. The developer shall conduct on-going evaluations of software development activities and the resulting software products to:

- a. Assure that each activity required by the contract or described in the CG Software Plan (CG-SP) is being performed in accordance with the contract and with the CG-SP.
- b. Assure that each software product required by this standard or by other contract provisions exists and has undergone software product evaluations, testing, and corrective action as required by this standard and by other contract provisions.

5.21.2 Software quality assurance records. The contractor shall record the results of software quality assurance activities. These records shall be maintained for the life of the contract. Problems in software products under project-level, or higher, configuration control and problems in activities required by the contract or described in the software development plan shall be handled as described in 5.17 (Corrective Action).

5.21.3 Independence in software quality assurance. The persons responsible for conducting software quality assurance evaluations shall not be the persons who developed the software product, performed the activity, or are responsible for the software product or activity. This does not preclude such persons from taking part in these evaluations. The persons responsible for assuring compliance with the contract shall have the resources, responsibility, authority, and organizational freedom to permit objective software assurance evaluations and to initiate and verify corrective actions.

6. NOTES

- 6.1 Intended use.** This standard is intended for software development as contracted for by the U.S. Coast Guard. Its application is described in 1.2.
- 6.2 Ordering software via CLIN.** The software product itself, usually in the form of executable code, should be ordered via Contract Line Item (CLIN) rather than via the Contract Data Requirement List (CDRL).
- 6.3 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) or the Automated Data Processing (ADP) equivalent, the CDRL shall identify the subset of the data identified below that are to be delivered. These data shall be developed as specified by an approved Data Item Description (DD Form 1664) or the Automated Data Processing (ADP) equivalent, and delivered in accordance with the approved CDRL incorporated into the contract. These data shall be delivered by the contractor in accordance with the contract or purchase order requirements. The DIDs related to this standard (CG-SDDS) are identified below. Copies of DIDs required by contractors in connection with specific acquisition functions should be obtained from the contracting officer.

| <u>Para.</u> | <u>Data Requirements Title</u> | <u>Applicable DID</u> |
|---------------------|--|--------------------------------|
| 5.1.1 | Coast Guard Software Plan (CG-SP) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-001 or DI-CGSDDS-007 |
| 5.1.2 5.1.3 | Coast Guard Software Test Document (CG-STD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-004 or DI-CGSDDS-007 |
| 5.1.4 | Coast Guard Software Plan (CG-SP) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-001 or DI-CGSDDS-007 |
| 5.1.5 | Coast Guard Software Plan (CG-SP) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-001 or DI-CGSDDS-007 |

Data requirements list and cross references

| <u>Para.</u> | <u>Data Requirements Title</u> | <u>Applicable DID</u> |
|----------------------------|---|-----------------------------------|
| 5.3.2 | Coast Guard SW Requirements Document (CG-SRD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-002 or DI-CGSDDS-007 |
| 5.3.3 | Coast Guard Software Design Document (CG-SDD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-002 or DI-CGSDDS-007 |
| 5.4.1 5.4.2 | Coast Guard Software Design Document (CG-SDD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-003 or DI-CGSDDS-007 |
| 5.5.1 | Coast Guard SW Requirements Document (CG-SRD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-002 or DI-CGSDDS-007 |
| 5.5.2 | Coast Guard SW Requirements Document (CG-SRD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-002 or DI-CGSDDS-007 |
| 5.6.1 5.6.2 5.6.3 | Coast Guard Software Design Document (CG-SDD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-003 or DI-CGSDDS-007 |
| 5.7.1 5.7.2 5.7.3 | Coast Guard Software Design Document (CG-SDD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-003 or DI-CGSDDS-007 |
| 5.10.1 5.10.2 5.10.6 | Coast Guard Software Test Document (CG-STD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-004 or DI-CGSDDS-007 |
| 5.11.1 5.11.2 5.11.6 | Coast Guard Software Test Document (CG-STD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-004 or DI-CGSDDS-007 |
| 5.12.1 5.12.2 5.12.6 | Coast Guard Software Test Document (CG-STD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-004 or DI-CGSDDS-007 |

Data requirements list and cross references cont.

| <u>Para.</u> | <u>Data Requirements Title</u> | <u>Applicable DID</u> |
|----------------------------|--|-----------------------------------|
| 5.13.6 | Coast Guard Software Test Document (CG-STD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-004 or DI-CGSDDS-007 |
| 5.14.1 | Coast Guard SW User/Operator Manual (CG-SUOM) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-005 or DI-CGSDDS-007 |
| 5.14.2 | Coast Guard SW User/Operator Manual (CG-SUOM) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-005 or DI-CGSDDS-007 |
| 5.14.3 | Coast SW User/Operator Manual (CG-SUOM)) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-005 or DI-CGSDDS-007 |
| 5.14.4 | Coast Guard SW User/Operator Manual (CG-SUOM) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-005 or DI-CGSDDS-007 |
| 5.14.5 | Coast Guard Software Support Document (CG-SSD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-006 or DI-CGSDDS-007 |
| 5.15.3 | Coast Guard Software Support Document (CG-SSD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-006 or DI-CGSDDS-007 |
| 5.15.4 | Coast Guard Software Support Document (CG-SSD) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-006 or DI-CGSDDS-007 |
| 5.21.1 5.21.2 5.21.3 | Coast Guard Software Plan (CG-SP) or Coast Guard Consolidated SW Devel. Document (CG-CSDD) | DI-CGSDDS-001 or DI-CGSDDS-007 |

Data requirements list and cross references cont.

- 6.4 Relationship between standard and CDRL.** If the CDRL calls for a data item different from the one named in corresponding paragraph(s) of this standard, the CDRL shall prevail and all references to the data item in the standard shall be interpreted to mean the data item in the CDRL.
- 6.5 Delivery of database/tool contents.** Unless otherwise specified in the contract, a CDRL may be satisfied by delivery of: 1) a repository or database containing the data specified in the cited DID; 2) a means of accessing that database or repository, such as a CASE tool, if not already available to the recipients designated on the CDRL; and 3) a hard-copy or electronically stored table of contents for the required data item, specifying how and where to access the data required in each paragraph of the DID.
- 6.6 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.
- 6.7 Subject term (key word) listing.**
- Acquisition
 - Architectural design
 - Baselines
 - Builds
 - Coding
 - Computer
 - Computer resources
 - Computer software
 - Computer software component
 - Computer software configuration item
 - Computer software unit
 - Configuration item
 - Configuration management
 - Database
 - Data item descriptions
 - Detailed design
 - Evolutionary development
 - Firmware
 - Incremental development
 - Non-developmental software
 - Operational concept
 - Process improvement
 - Requirements analysis
 - Reusable software

Risk management
Security
Software
Software development
Software development file
Software development library
Software development planning
Software development process
Software engineering
Software engineering
environment
Software manuals
Software product evaluation
Software requirements analysis
Software safety
Software support
Software test environment
Software testing
Software version
System design
System requirements analysis
System testing
Tailoring
Testing

APPENDIX A

LIST OF ACRONYMS AND ABBREVIATIONS

A.1 Purpose. This appendix provides a list of all acronyms and abbreviations used in this standard, with their associated meanings. This appendix is not a mandatory part of the standard. The material contained in this appendix is for information only.

A.2 Acronyms.

| | |
|---------|--|
| CG-SDD | Coast Guard Software Design Document |
| CG-SP | Coast Guard Software Plan |
| CG-SRD | Coast Guard Software Requirements Document |
| CG-SSD | Coast Guard Software Support Document |
| CG-STD | Coast Guard Software Test Document |
| CG-SUOM | Coast Guard Software User/Operator Manual |
| CG-CSDD | Coast Guard Consolidated Software Development Document |
| CG-PR | Coast Guard Progress Report |
| CASE | Computer Aided Software Engineering, Computer Aided System Engineering |
| CDRL | Contract Data Requirements List |
| CIDS | Critical Item Development Specification |
| CLIN | Contract Line Item |
| CM | Configuration Management |
| COTS | Commercial Off-The-Shelf |
| CSCI | Computer Software Configuration Item |
| CSOM | Computer System Operator Manual |
| DBDD | Database Design Document |
| DBMS | Database Management System |
| DID | Data Item Description |
| ECP | Engineering Change Proposal |
| FAR | Federal Acquisition Regulation |
| FCA | Functional Configuration Audit |
| GFS | Government Furnished Software |
| HOL | High Order Language |
| HWCI | Hardware Configuration Item |
| IDD | Interface Design Document |
| I/O | Input/Output |
| IRS | Interface Requirements Specification |
| IV&V | Independent Verification and Validation |
| NDI | Non-developmental Item |
| NDS | Non-developmental Software |
| PCA | Physical Configuration Audit |
| PIDS | Prime Item Development Specification |
| SCN | Specification Change Notice |
| SDF | Software Development File |
| SDL | Software Development Library |

SOW Statement of Work
SW Software
WBS Work Breakdown Structure

APPENDIX B

REQUIREMENTS FOR SOFTWARE CODING STANDARDS

- B.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, subject to tailoring by the contracting agency.
- B.2 Requirements for coding standards.** The contractor shall implement software coding standards that comply with the following requirements.
- B.2.1 Presentation style.** The coding standards shall describe rules and conventions for the format of the source code, which may include paper listings, listings stored on electronic media, or both. The rules and conventions for presentation style shall include standards for:
- a. Indentation and spacing
 - b. Use of capitalization
 - c. Uniform presentation of information (e.g., grouping together all data declarations)
 - d. Layout of source code listings
 - e. Conditions under which comments will be provided and the format to be used
 - f. Size of code aggregates
 - g. Use of headers to document the code, including, for example:
 - 1) Name and project-unique identifier of the code entity
 - 2) Name and project-unique identifier of software unit(s) using the entity
 - 3) Version identifier
 - 4) File name/directory/library
 - 5) Purpose
 - 6) Reference to requirements implemented
 - 7) Names of designer and programmer
 - 8) Date/time of last compilation
 - 9) Input
 - 10) Outputs
 - 11) Global variables set or used
 - 12) Major data structures
 - 13) Data conversions performed
 - 14) Algorithms used
 - 15) Calling routines
 - 16) Called routines
 - 17) Assumptions, constraints, limitations, side effects, unusual features

18) Modification history, including, for example, for each change:

- B.2.2 Naming.** The coding standards shall describe rules and conventions governing the selection of identifiers used in the source code listings (e.g., identifiers for variables, parameters, packages, procedures, units, and other aggregates of code). Restrictions on the use of reserved words and keywords shall be identified.
- B.2.3 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.
- B.2.4 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.
- B.2.5 Complexity.** The coding standards shall describe controls and restrictions on the complexity of code aggregates.

APPENDIX C

CATEGORY AND PRIORITY CLASSIFICATIONS FOR PROBLEM REPORTING

- C.1 Purpose.** This appendix contains requirements for a category and priority classification scheme to be applied to all problems detected in software product evaluations and in CSCI testing, CSCI integration testing, and system testing. This appendix is a mandatory part of this standard, subject to the following conditions: 1) these requirements may be tailored by the contracting agency, and 2) the contractor may use alternate category and priority schemes if approved by the contracting agency.
- C.2 Classification by category.** The contractor shall assign each problem to one or more of the categories in Figure 2.
- C.3 Classification by priority.** The contractor shall assign each problem one of the priorities in Figure 3.

| CATEGORY | APPLIES IF A PROBLEM HAS BEEN FOUND IN: |
|-----------------|--|
| a. Planning | One of the plans developed for the project |
| b. Concept | The operational concept |
| c. Requirements | The system or software requirements |
| d. Design | The design of the system of software |
| e. Software | The coded software |
| f. Testing | The test plans, cases, test procedures, or test report |
| g. Manuals | The user or support manuals |
| h. Process | The process being followed on the project |
| i. Other | Any other aspect of the project |

FIGURE 2. Categories to be used for classifying problems.

FIGURE 2. Categories to be used for classifying problems.

| PRIORITY | APPLIES IF A PROBLEM COULD RESULT IN ONE OR MORE OF THESE IMPACTS |
|----------|---|
| 1 | <ul style="list-style-type: none"> a. Prevent the accomplishment of an operational or mission essential capability b. Jeopardize safety c. Cause significant technical, cost , or schedule risks to the project or to life cycle support of the system |
| 2 | <ul style="list-style-type: none"> a. Adversely affect the accomplishment of an operational or mission essential capability and no work-around solution is known b. Adversely affect technical, cost, or schedule risks to the project or to life cycle support of the system, and no work-around is known |
| 3 | <ul style="list-style-type: none"> a. Adversely affect the accomplishment of an operational or mission essential capability but a work-around solution is known b. Adversely affect technical, cost, or schedule risks to the project or to life cycle support of the system, but a work-around is known |
| 4 | <ul style="list-style-type: none"> a. Result in user/operator inconvenience or annoyance but does not affect a required operational or mission essential capability b. Result in inconvenience or annoyance for development or support personnel, but does not prevent the accomplishment of those responsibilities |
| 5 | Any other effect |

Figure 3. Priorities to be used for classifying problems

Figure 3. Priorities to be used for classifying problems

APPENDIX D

SOFTWARE PRODUCT EVALUATIONS

- D.1 Purpose.** This appendix identifies the software development products that are to undergo software product evaluations, identifies the criteria to be used for each evaluation, and contains a default set of definitions for the evaluation criteria. This appendix is a mandatory part of this standard, subject to the following conditions: 1) these requirements may be tailored by the contracting agency, 2) the contractor may use alternate criteria or definitions if approved by the contracting agency, and 3) if the development of a given product has been tailored out of the standard, the requirement to evaluate that product does not apply.
- D.2 Required evaluations.** Figure 4 identifies the software development products that are to undergo software product evaluations and states the criteria to be applied to each one. Each product and criterion is labeled for purposes of identification and tailoring. For convenience, they may be treated as subparagraphs of this paragraph (referring to the first criterion, for example, D.2.1.a). The products are expressed in lower case letters to convey generic products, not necessarily in the form of hardcopy documents. Evaluations of system level products are to be interpreted as participation in these evaluations, as interpreted in paragraph 1.2.5. Some of the criteria are subjective. Because of this, there is no requirement to prove that the criteria have been met; the requirement is to perform the evaluations using these criteria and to identify possible problems for discussion and resolution. Definitions for the criteria are in D.3.
- D.3 Criteria definitions.** The following paragraphs provide definitions for the criteria in Figure 4 that may not be self-explanatory. The criteria are listed in alphabetical order, matching as closely as possible the wording used in Figure 4.
- D.3.1 Accurately describes (an item).** This criterion, applied to user/operator/programmer instructions and to the "as-built" design and version descriptions, means that the instructions or descriptions are correct depiction of the software or other item described.
- D.3.2 Adequate (unit/unit integration) test cases, procedures, results.** Test cases are adequate if they cover all applicable requirements, include both nominal and stress tests, and specify the

inputs to be used, the expected results, and the criteria to be used for evaluating those results. Test procedures are adequate if they specify the steps to be followed in carrying out each test case. Test results are adequate if they describe the results of all test cases and show that all criteria have been met, possibly after revision and retesting of the unit or group of units.

D.3.3 Consistent with indicated products(s). This criterion means that: (1) no statement or representation in one product contradicts a statement or representation in the other products, (2) a given term, acronym, or abbreviation means the same thing in all of the products, and (3) a given item or concept is referred to by the same name or description in all of the products.

D.3.4 Contains all applicable information in (a specified DID). This criterion uses the DIDs to specify the required content of intermediate and final software development products, regardless of whether a deliverable document has been ordered. Allowances are to be made for the applicability of each DID topic. The formatting specified in the DID (required paragraphing and numbering) are not relevant to this evaluation.

D.3.5 Covers (a given set of items). A software development product "covers" a given set of items if every item in the set has been dealt with in the product. For example, a plan covers the SOW if every provision in the SOW is included in the plan; a design covers a set of requirements if every requirement has been incorporated/dealt with in the design; a test plan covers a set of requirements if every requirement is the subject of one or more tests.

D.3.6 Feasible. This criterion means that, in the knowledge and experience of the evaluator, a given concept, set of requirements, design, test, etc. violates no known principles or lessons learned that would render it impossible to carry out.

D.3.7 Follows software development plan. This criterion means that the product shows evidence of having been developed in accordance with the approach described in the software development plan. Examples include following design and coding standards described in the plan. For the software development plan itself, this criterion applies to updates to the initial plan.

D.3.8 Includes both nominal and stress tests. This criterion means that a set of planned tests or test cases includes both normal/expected test inputs and

conditions, and test inputs and conditions at the limits of the item's specified requirements. If an item's requirements include required behavior in response to illegal or out-of-bounds conditions, such illegal or out-of-bounds conditions fall within the definition of stress tests.

- D.3.9 **Internally consistent.** This criterion means that: (1) no two statements or representations in a product contradict one another, (2) a given term, acronym, or abbreviation means the same thing throughout the product, and (3) a given item or concept is referred to by the same name or description throughout the product.
- D.3.10 **Meets CDRL, if applicable.** This criterion applies if the product being evaluated is specified in the CDRL and has been formatted for delivery at the time of evaluation. It focuses on the format, markings, and other provisions specified in the CDRL, rather than on content, covered by other criteria.
- D.3.11 **Meets SOW, if applicable.** This criterion means that the product complies with any Statement of Work provisions regarding it. For example, the Statement of Work may place constraints on the operational concept or the design.
- D.3.12 **Presents a sound approach.** This criterion means that, based on the knowledge and experience of the evaluator, a given plan represents a reasonable way to carry out the required activities.
- D.3.13 **Shows evidence that (an item under test) meets its requirements.** This criterion means that recorded test results show that the item under test either passed all tests the first time or was revised and retested until the tests were passed.
- D.3.14 **Testable.** A requirement or set of requirements is considered to be testable if an objective and feasible test can be designed to determine whether each requirement is met by the software.
- D.3.15 **Understandable.** This criterion means "understandable by the intended audience." For example, products intended for programmer-to-programmer communication need not be understandable by non-programmers. A product that correctly identifies its audience (based on information in Block 3 of the corresponding DID) and is considered understandable to that audience meets this criterion.
- D.3.16 **Well-reasoned use of reusable components.** This criterion means there is evidence that reusable components have been considered, sound analysis has been carried out to determine their

suitability, and sound conclusions have been reached regarding their use.

| Evaluation Criteria | | | | | | | |
|---------------------------------------|--------------------------------------|------------------------------|-------------------------------|-----------------------|------------------------------|-----------------------------|--|
| Software product | Contains all applic. info in: | Meets SOW, if applic. | Meets CDRL, if applic. | Understandable | Internally Consistent | Follows SW dev. Plan | Additional Criteria |
| 1. Software development plan (5.1.1) | a. CG-SP DID | b. | c. | d. | e. | f. (updates) | g. Covers all activities/deliverables in SOW and CDRL h. Consistent with other project plans i. Presents a sound approach to the development |
| 2. Software test plan (5.1.2, 5.1.3) | a. CG-STP DID | b. | c. | d. | e. | f. | g. Covers all software-related qualification activities in the SOW h. Covers all requirements for the items under test i. Consistent with other project plans j. Presents a sound approach to testing |
| 3. Software installation plan (5.1.4) | a. CG-SP DID | b. | c. | d. | e. | f. | g. Covers all user site installation activities in the SOW h. Consistent with other project plans i. Presents a sound approach to the installation |
| 4. Software transition plan (5.1.5) | a. CG-SP DID | b. | c. | d. | e. | f. | g. Covers all transition-related activities in the SOW h. Consistent with other project plans i. Presents a sound approach to transition |
| 5. Operational concept (5.3.2) | a. CG-SRD DID | b. | c. | d. | e. | f. | g. Feasible |
| 6. System requirements (5.3.3) | a. CG-SRD DID | b. | c. | d. | e. | f. | a. Covers the operational concept b. Feasible c. Testable |

FIGURE 4. Software products and associated evaluation criteria

FIGURE 4. Software products and associated evaluation criteria

| Evaluation Criteria | | | | | | | |
|--|--------------------------------------|------------------------------|-------------------------------|-----------------------|------------------------------|-----------------------------|---|
| Software product | Contains all applic. info in: | Meets SOW, if applic. | Meets CDRL, if applic. | Understandable | Internally Consistent | Follows SW dev. Plan | Additional Criteria |
| 7. System-wide design decisions (5.4.1) | a. CG-SDD DID | b. | c. | d. | e. | f. | g. Consistent with system requirements h. Feasible |
| 8. System architectural design (5.4.2) | a. CG-SDD DID | b. | c. | d. | e. | f. | g. Covers the system requirements h. Consistent with system-wide design decisions i. Feasible |
| 9. CSCI requirements (5.5) | a. CG-SRD DID | b. | c. | d. | e. | f. | g. Covers system requirements allocated to the CSCI h. Feasible i. Testable |
| 10. CSCI-wide design decisions (5.6.1) | a. CG-SDD DID | b. | c. | d. | e. | f. | g. Consistent with CSCI requirements h. Feasible |
| 11. CSCI architectural design (5.6.2) | a. CG-SDD DID | b. | c. | d. | e. | f. | g. Covers CSCI requirements h. Consistent with CSCI-wide design decisions i. Feasible |
| 12. CSCI detailed design (5.6.3) | a. CG-SDD DID | b. | c. | d. | e. | f. | g. Covers CSCI requirements allocated to each unit h. Consistent with CSCI-wide design decisions |
| 13. Implemented Software (5.7.1) | a. CG-SDD DID | b. | c. | d. | e. | f. | g. Covers the CSCI detailed design. |
| 14. CSCI qualification test descriptions (5.9.1) | a. CG-STD DID | b. | c. | d. | e. | f. | g. Covers all CSCI requirements. |

FIGURE 4. Software products and associated evaluation criteria - continued.
Evaluation Criteria

FIGURE 4. Software products and associated evaluation criteria - continued.

| Evaluation Criteria | | | | | | | |
|---|--------------------------------------|------------------------------|-------------------------------|-----------------------|------------------------------|-----------------------------|---|
| Software product | Contains all applic. info in: | Meets SOW, if applic. | Meets CDRL, if applic. | Understandable | Internally Consistent | Follows SW dev. Plan | Additional Criteria |
| 15. CSCI qualification test results (5.10) | a. CG-STD DID | b. | c. | d. | e. | f. | g. Covers all planned CSCI qualification test cases h. Shows evidence that the CSCI meets its requirements |
| 16. System qualification test descriptions (5.12.1) | a. CG-STD DID | b. | c. | d. | e. | f. | g. Covers all system requirements |
| 17. System qualification test results (5.12) | a. CG-STD DID | b. | c. | d. | e. | f. | g. Covers all planned system qualification test cases. h. Shows evidence that the system meets its requirements |
| 18. Executable software (5.15.1) | a. CG-SSD DID | b. | c. | d. | e. | f. | g. Meets delivery requirements h. All software necessary for execution is present i. Version exactly matched version that passed testing j. Deliverable media accurately labeled |
| 19. Software version descriptions (5.15.4) | a. CG-SSD DID | b. | c. | d. | e. | f. | g. Accurately identifies the version of each software component (file, unit, CSCI, etc) delivered h. Accurately identifies the changes incorporated |
| 20. Software user manuals (5.14.1) | a. CG-SUOM DID | b. | c. | d. | e. | f. | g. Accurately describes software installation and use to the intended audience of this manual |
| 21. Software input/output manuals | a. CG-SUOM DID | b. | c. | d. | e. | f. | g. Accurately describes software input/output to the intended audience of this manual |

FIGURE 4. Software products and associated evaluation criteria - continued.
Evaluation Criteria

| | | | | | | | |
|----------|--|--|--|--|--|--|--|
| (5.14.3) | | | | | | | |
|----------|--|--|--|--|--|--|--|

FIGURE 4. Software products and associated evaluation criteria - continued.

| Evaluation Criteria | | | | | | | |
|---|--------------------------------------|------------------------------|-------------------------------|-----------------------|------------------------------|-----------------------------|--|
| Software product | Contains all applic. info in: | Meets SOW, if applic. | Meets CDRL, if applic. | Understandable | Internally Consistent | Follows SW dev. Plan | Additional Criteria |
| 22. Software center operator manuals (5.14.2) | a. CG-SUOM DID | b. | c. | d. | e. | f. | g. Accurately describes software installation and operation to the intended audience of this manual |
| 23. Computer operation manuals (5.14.4) | a. CG-SUOM DID | b. | c. | d. | e. | f. | g. Accurately describes the operational characteristics of the computer |
| 24. Source files (5.14.2) | a. CG-SUOM DID | b. | c. | d. | e. | f. | g. Meets delivery requirements h. All required software is present i. Version exactly matches version that passed testing j. Delivery media accurately labeled |
| 25. "As built" CSCI design and related information (5.14.4) | a. CG-SUOM DID | b. | c. | d. | e. | f. | g. Accurately describes the "as built" design of the CSCI h. Accurately describes compilation/build procedures i. Accurately describes modification procedures j. Source files cover all units in the CSCI design. k. Measured resource utilization meets CSCI requirements. |
| 26. "As built" system design (5.14.5) | a. CG-SSD DID | b. | c. | d. | e. | f. | g. Accurately describes the "as built" system design |

FIGURE 4. Software products and associated evaluation criteria - continued.

FIGURE 4. Software products and associated evaluation criteria - continued.

APPENDIX E

CANDIDATE JOINT MANAGEMENT REVIEWS

- E.1 Purpose.** This appendix describes a candidate set of joint management reviews that might be held during a software development effort (see 5.19). This appendix is a not a mandatory part of this standard.
- E.2 Assumptions.** This appendix makes the following assumptions:
- a. The Government has reviewed the subject products in advance, and one or more joint technical reviews have been held to resolve issues, leaving the joint management review as a forum to resolve open issues and reach agreement as to the acceptability of each product.
 - b. Any of the reviews may be conducted incrementally, dealing at each review with a subset of the listed items or a subset of the system or CSCI(s) being reviewed.
- E.3 Candidate reviews.** Given below are a set of candidate joint management reviews that might be held during a software development project. There is no intent to preclude alternatives or combinations of these reviews. Objectives for joint management reviews are given in 5.19.
- E.3.1 Software plan reviews.** These reviews are held to resolve open issues regarding one or more of:
- a. The software development plan -- (CG-SP) DID
 - b. The software test plan -- (CG-STD)
 - c. The software support plan -- (CG-SP) DID
 - d. The software installation plan -- (CG-SP) DID
- E.3.2 Software operational concept reviews.** These reviews are held to resolve open issues regarding the operational concept for a software system.
- E.3.3 Software system requirements reviews.** These reviews are held to resolve open issues regarding the specified requirements for a software system.
- E.3.4 Software system design reviews.** These reviews are held to resolve open issues regarding one or more of:

- a. The behavioral design of a software system
 - b. The architectural design of a software system
- E.3.5 Software requirements reviews.** These reviews are held to resolve open issues regarding one or more of:
- a. The engineering requirements for a CSCI
 - b. The interface requirements for a CSCI
- E.3.6 Software design reviews.** These reviews are held to resolve open issues regarding one or more of:
- a. The behavioral design of a CSCI
 - b. The architectural design of a CSCI
 - c. The detailed design of a CSCI
 - d. The design of CSCI interfaces
 - e. The design of databases
- E.3.7 Software test readiness reviews.** These reviews are held to resolve open issues regarding one or more of:
- a. The status of the software test environment
 - b. The CSCI, CSCI integration, and/or system test cases and test procedures to be used
 - c. The status of the software and databases in preparation for testing
 - d. Readiness of software manuals for use and validation in testing
- E.3.8 Software supportability reviews.** These reviews are held to resolve open issues regarding one or more of:
- a. The adequacy of the software product specifications to serve as the basis for supporting the software
 - b. The supportability of the source code
 - c. The status of software support manuals
 - d. The status of transition activities
- E.3.9 Software usability reviews.** These reviews are held to resolve open issues regarding one or more of:
- a. User and operator manuals
 - b. The readiness of the software and databases for installation
 - c. The status of preparations for installation and training
 - d. The status of on-going installation and training activities

APPENDIX F

CANDIDATE MANAGEMENT INDICATORS

- F.1 Purpose.** This appendix identifies a set of management indicators that might be used on a software development project (see 5.20.6). This appendix is not a mandatory part of this standard.
- F.2 Candidate indicators.** Given below are a set of candidate management indicators that might be used on a software development project. There is no intent to impose these indicators or to preclude others.
- a. Requirements volatility: total number of requirements and requirement changes over time.
 - b. Software size: number of units or lines or code, planned and actual, over time.
 - c. Software staffing: number of planned vs actual staff over time.
 - d. Complexity of each software unit.
 - e. Software progress: planned and actual number of software units designed, coded, unit tested, and integrated over time.
 - f. Problem change/report status: total number, number closed, number opened in the current reporting period, age, priority.
 - g. Build release content: planned and actual number of software units released in each build.
 - h. Computer resource utilization: planned and actual use of total memory, input/output resources, processor throughput over time.
 - i. Milestone performance: planned and actual dates of key project milestones.

APPENDIX G

APPLYING CG-SDDS TO LIFE CYCLE MODELS

- G.1 Purpose.** This appendix describes candidate software life cycle models and provides guidance on applying CG-SDDS to these models. Selecting a life cycle model for the system and software is part of the Government's acquisition strategy and is usually performed by the Government. This appendix provides guidance to the Government and contractor in understanding these models and how CG-SDDS can be applied to each. This appendix is not a mandatory part of the standard.
- G.2 Candidate life cycle models.** There are many possible life cycle modes. Three examples are identified below. Their similarities and differences are summarized in Figure 5.
- a. Grand design. The "grand design" model is essentially a "once-through," do each step once model. Simplistically: determine user needs, define requirements, design the system, develop the system, test, fix, and deliver.
 - b. Incremental. The "incremental" model determines user needs and defines the system requirements, then performs the rest of the development in a sequence of builds. The first build incorporates part of the planned capabilities, the next build adds more capabilities, and so on, until the system is complete.
 - c. Evolutionary. The evolutionary model also develops a system in builds, but differs from the incremental model in acknowledging that the user need is not fully understood and all requirements cannot be defined up front. In this model, user needs and system requirements are partially defined up front, then are refined in each succeeding build.

| Grand Design | | Incremental | | Evolutionary | |
|---|-------------------|---|-------------------|--|-------------------|
| Risk Item (Reasons against this strategy) | Risk Level | Risk Item (Reasons against this strategy) | Risk Level | Risk Item (Reasons against this strategy) | Risk Level |
| Requirements are not well understood | H | Requirements are not well understood | H | User prefers all capabilities at first delivery | M |
| System too large to do all at once | M | User prefers all capabilities at first delivery | M | | |
| Rapid changes in mission technology anticipated - may change requirements | H | Rapid changes in mission technology anticipated - may change requirements | H | | |
| Limited staff or budget available now | M | | | | |
| Opportunity Item (Reasons to use this strategy) | Opp Level | Opportunity Item (Reasons to use this strategy) | Opp Level | Opportunity Item (Reasons to use this strategy) | Opp Level |
| User prefers all capabilities at first delivery | M | Each capability is needed | H | Each capability is needed | H |
| User prefers to phase out old system all at once | L | System breaks naturally into increments | M | System breaks naturally into increments | M |
| | | Funding/staffing will be incremental | H | Funding/staffing will be incremental | H |
| | | | | User feedback and monitoring of technology changes is needed to understand full requirements | H |
| | | | | DECISION: USE THIS STRATEGY | |

FIGURE 6. Sample risk analysis for determining the appropriate program strategy.

FIGURE 5. Key features of selected life cycle models.

- G.3 Selecting an appropriate model.** The appropriate life cycle model for a given system depends upon system characteristics. Figure 6 illustrates a risk analysis approach for selecting an appropriate model for a given system. The approach consists of listing risk items (negatives) and opportunity items (positives) for each model; ranking each item High, Medium, or Low; and making a decision on which model to use based on a trade-off among the items and their ranks. The fill-ins shown are sample considerations only. An actual analysis may use others. The "USE" entry on the bottom line shows which model was selected.
- G.4 Planning the builds for an incremental or evolutionary model.** If the selected life cycle model is incremental or evolutionary, it is necessary to lay out a plan for what parts of the software will be developed in each build. Figure 7 provides an example of such a plan for an incremental model. In the example, there are two planned CSCIs, each to be defined and developed in builds. Portions of both CSCI 1 and CSCI 2 are to be developed in the first Build. The remainder of the system is to be developed in Build 2. As shown in the lower part of the figure, the requirements for these components are drafted and refined over the builds, beginning before Build 1 and continuing through Build 2. Such a plan may be developed by the Government or proposed by the contractor.
- G.5 Applying CG-SDDS to each life cycle model.** Paragraph 4.1 of this standard requires the contractor to map the activities of this standard onto the life cycle model established for the software. This involves: 1) deciding which CG-SDDS activities apply in each build, and 2) laying out a schedule of the activities selected for each build.
- G.5.1 Determining which activities apply in each build.** Each of the life cycle models described above may be thought of as consisting of builds. The "grand design" model has a single build; the incremental and evolutionary models have multiple builds. In determining which activities apply to each build, the key question to be answered is: what are the objectives of the build? The planning illustrated in Figure 7 is the starting point for this determination. Figure 8 provides an example of a worksheet for more detailed planning. This worksheet provides a set of candidate objectives for each build and identifies the CG-SDDS paragraphs that should be applied to meet those objectives. The following guidelines apply:

- a. Each worksheet applies to a given build and a given subset of the software (a CSCI, part of a CSCI, multiple CSCIs, software of a given type, etc.). A complete set of worksheets is needed to fully define a project, even when there is only one build.
- b. If early builds are devoted to experimentation, developing "throw-away" software to arrive at a system concept or system requirements, it may be appropriate to forgo certain formalities, such as coding standards, that will be imposed later on the "real" software. If the early software will be used later, such formalities may be appropriate from the start. Each project must determine this balance.

G.5.2 Scheduling the selected activities in each build. The second task in mapping the activities of CG-SDDS to the selected life cycle model is scheduling the activities in each build. The following guidelines apply:

- a. Different CSCIs may be on different schedules. A common mistake is to treat all CSCIs as though they must be developed in "lock-step," reaching key milestones at the same time.
- b. Different units within a given CSCI may be on different schedules. As with CSCIs, it is a common mistake to treat units as though they must be developed in "lock-step."
- c. The activities in CG-SDDS need not be performed sequentially. Several may be taking place at one time, and an activity may be performed continually or intermittently throughout a build or over multiple builds. The activities in each build should be laid out in the manner that best suits the work to be done.

| Grand Design | | Incremental | | Evolutionary | |
|---|-------------------|---|-------------------|--|-------------------|
| Risk Item (Reasons against this strategy) | Risk Level | Risk Item (Reasons against this strategy) | Risk Level | Risk Item (Reasons against this strategy) | Risk Level |
| Requirements are not well understood | H | Requirements are not well understood | H | User prefers all capabilities at first delivery | M |
| System too large to do all at once | M | User prefers all capabilities at first delivery | M | | |
| Rapid changes in mission technology anticipated - may change requirements | H | Rapid changes in mission technology anticipated - may change requirements | H | | |
| Limited staff or budget available now | M | | | | |
| Opportunity Item (Reasons to use this strategy) | Opp Level | Opportunity Item (Reasons to use this strategy) | Opp Level | Opportunity Item (Reasons to use this strategy) | Opp Level |
| User prefers all capabilities at first delivery | M | Each capability is needed | H | Each capability is needed | H |
| User prefers to phase out old system all at once | L | System breaks naturally into increments | M | System breaks naturally into increments | M |
| | | Funding/staffing will be incremental | H | Funding/staffing will be incremental | H |
| | | | | User feedback and monitoring of technology changes is needed to understand full requirements | H |
| | | | | DECISION: USE THIS STRATEGY | |

FIGURE 6. Sample risk analysis for determining the appropriate program strategy.

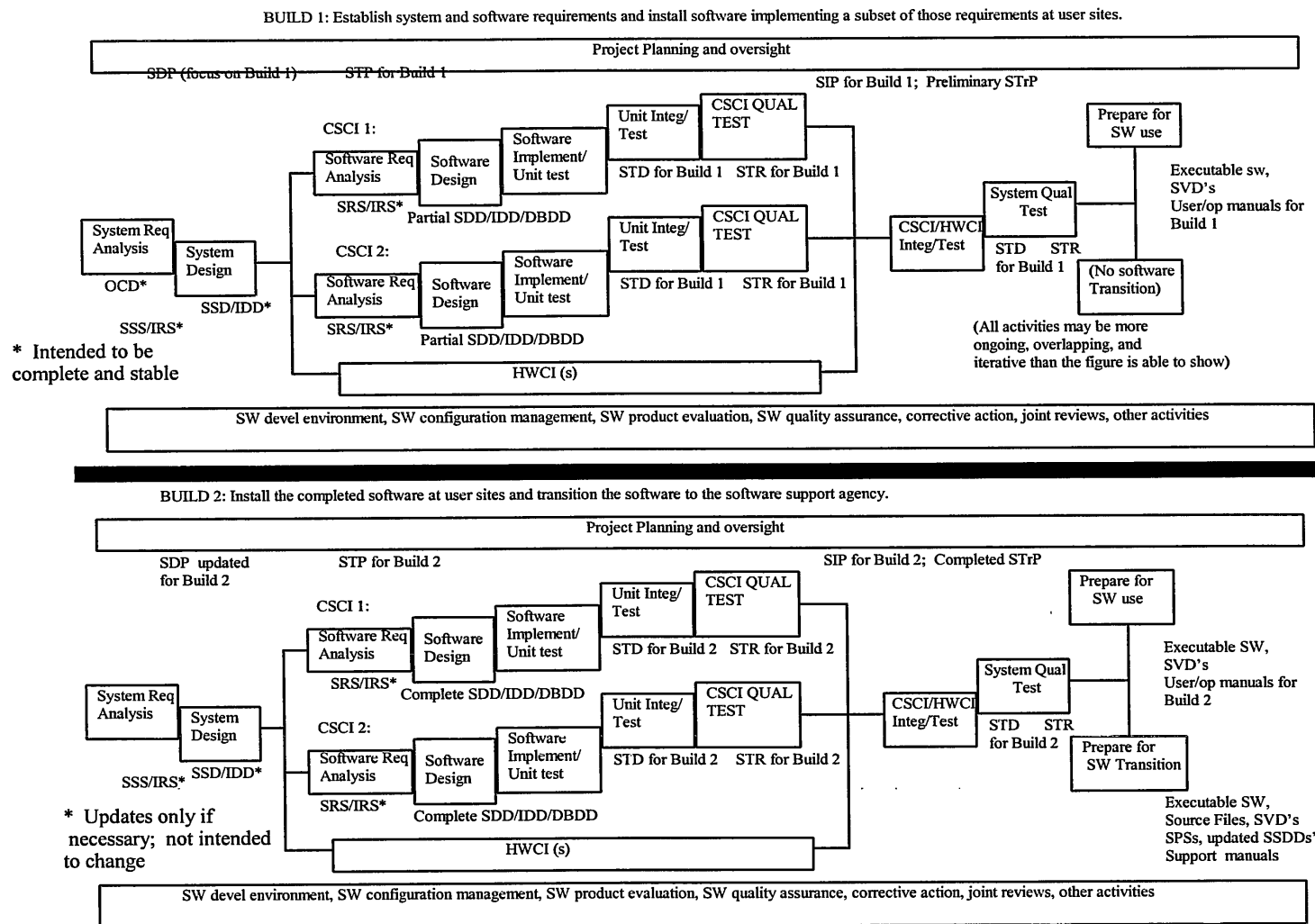


FIGURE 7. One Possible Way of applying CG-SDDS to the Incremental program strategy

Figure 7. One possible way of applying CD-SDDS to the Incremental Program Strategy.

| BUILD PLANNING WORKSHEET | | BUILD | | | |
|---|--|---|---|---|--|
| | | 1. | 2. | 3. | 4. |
| 1. Identify at right the objectives of each build | | Deliver to selected users an operational prototype that meets the following system-level requirements: SSS-1, SSS-5,... SSS-1250* | Deliver to selected users an operational prototype that meets the requirements of Build 1, plus SSS-2, SS-3, SSS-15,... SSS-1249* | Deliver to all users a tested system that meets the requirements of Build 1 and 2, plus, SSS-4, SSS-7, SSS-10...SSS-1248* | Deliver to all users a tested system that meets all the system-level requirements: transition to designated support agency |
| 2. Indicate below which activities are to be accomplished during the development of each build. Add clarifying notes as needed. | | | | | |
| Para | Activity | | | | |
| 5.1 | PROJECT PLANNING AND OVERSIGHT | | | | |
| 5.1.1 | Plan the software development effort | Yes: Plan Build 1 in detail; Builds 2-4 in general | Yes: Plan Build 2 in detail; Builds 3-4 in general | Yes: Plan Build 3 in detail; Build 4 in general | Yes: Plan Build 4 in detail; |
| 5.1.2 | Plan for CSCI qualification testing | No: No CSCI qual testing in this build | No: No CSCI qual testing in this build | Yes: Plan for CSCI qual testing in this build | Yes: Update for CSCI qual testing in this build |
| 5.1.3 | Plan for system qualification testing | No: No system qual testing in this build | No: No system qual testing in this build | Yes: Plan for system qual testing in this build | Yes: Update for system qual testing in this build |
| 5.1.4 | Plan for installing software at user sites | No: let users install on their own | No: let users install on their own | Yes: Plan to install at users site | Yes: Update as needed for installation of Build 4 |
| 5.1.5 | Plan for transitioning software to the support agency | Yes: Very Preliminary planning only | Yes: Update Preliminary plans | Yes: Update Preliminary plans | Yes: Finalize transition planning |
| 5.1.6 | Follow plans: perform management review | Yes: For those plans that are in effect | Yes: For those plans that are in effect | Yes: For those plans that are in effect | Yes: For those plans that are in effect |
| 5.2 | ESTABLISHING A SOFTWARE DEVELOPMENT ENVIRONMENT | | | | |
| 5.2.1 | Establish a software engineering environment | Yes: As needed for Build 1 | Yes: Update as needed for Build 2 | Yes: Update as needed for Build 3 | Yes: Update as needed for Build 4 |
| 5.2.2 | Establish a software test environment | Yes: As needed for Build 1 testing | Yes: As needed for Build 2 testing | Yes: Set up fully for Build 3 qualification testing | Yes: Update as needed for Build 4 qualification testing |

FIGURE 8. Example of a Build Planning worksheet.

* Refers to System/Subsystem Specifications

FIGURE 8. Example of a Build Planning worksheet.

* Refers to System/Subsystem Specifications

APPENDIX H

GUIDANCE ON DELIVERABLES

- H.1 Purpose.** This appendix provides guidance on the deliverables to be required on a software development project. This appendix is not a mandatory part of this standard.
- H.2 Ordering deliverables.** CG-SDDS has been worded to differentiate between the planning/engineering activities that make up a software development project and the generation of deliverables. A key objective of this wording is to eliminate the notion that the Government must order a given deliverable in order to have planning or engineering work take place. Under CG-SDDS, the planning and engineering work takes place regardless of what deliverables are ordered, unless a given activity is tailored out of the standard. In addition, joint technical reviews have been included to review the results of that work in its natural form, without the generation of deliverables. Deliverables should be ordered only when there is a genuine need to have planning or engineering information transformed into a deliverable, recognizing that this transformation requires time and effort that would otherwise be spent on the engineering effort.
- H.3 Scheduling deliverables.** CG-SDDS has been structured to support a variety of life cycle models and to provide the contractor flexibility in laying out a software development process that will best suit the work to be done. All of this flexibility can be canceled by rigid scheduling of deliverables on the CDRL. If the CDRL lays out a strict "waterfall" sequence of deliverables, little room is left to propose innovative development processes. If the CDRL forces all CSCIs into lock-step with each other, little room is left to develop the CSCIs in an optimum order. To the maximum extent possible, the CDRL should avoid such predetermination, leaving the door open for incremental delivery of products, staggered development of CSCIs, and other variations to optimize the software development effort. The contractor's software development plan will lay out a proposed schedule that meets the constraints in the CDRL. Final agreement on scheduling can take place at that time.
- H.4 Format of deliverables.** Traditional deliverables take the form of paper documents exactly following DID formats. While this form works well for some things, it is not the only form, and alternatives should be considered. One variation from paper

documents is word processing files containing those documents. This format saves paper, but still requires the contractor to format the information as required by the DID. Another variation is specifying that a paper or word processor document is to include all DID contents but may be in the contractor's format. Yet another variation is allowing deliverables to take forms that are not traditional documents at all, such as data in computer-aided software engineering (CASE) tools. These variations in required format can be specified on the CDRL, minimizing the time spent transforming actual work products into deliverables.

APPENDIX I

ACTIVITIES AND DELIVERABLES FOR REUSABLE SOFTWARE

I.1 Purpose. This appendix provides guidance on interpreting CG-SDDS when applied to reusable software being incorporated into software to which CG-SDDS is applied. This appendix makes no statement about activities or deliverables for software being generated for the purpose of being reusable. The latter subject is currently under study in DoD and is out of scope for CG-SDDS. This appendix is a not a mandatory part of this standard.

I.2 Interpreting CG-SDDS activities for reusable software.

Figure 11 provides guidance on interpreting CG-SDDS activities when applied to reusable software. It identifies five possible cases and maps these cases against the activities in CG-SDDS, providing an interpretation of each activity for each case. Key issues in this mapping are whether the software is going to be used unmodified or will be modified; whether the software constitutes an entire CSCI or a software unit; and whether or not the CSCI or unit has an established, positive performance record (no firm criteria exist for making this determination; it is a judgment call). Note that the figure is presented in a conditional manner: If an activity in the left column is required for a piece of software, the figure tells how to interpret the activity for the reusable components of that software

| Interpret the activity as follows for each type of existing, reusable software | | | | | |
|---|---|---|--|---------------------------------------|---|
| If this CG SDDS activity is required: | For CSCI's to be used unmodified | | For software units to be used unmodified | | For software units being modified for/during project |
| | Positive Performance Record | No or Poor Performance Record | Positive performance record | No or Poor Performance Record | |
| 5.1 Project Planning | Include the activities in this figure in project plans | | | | |
| 5.2 Establishing a software development environment | Establish and apply a software test environment, software development library, and software development files as appropriate to perform the activities in this figure | | | | Apply full requirements |
| 5.3 | Consider software's capabilities in defining the operational concept & system requirements | | | | |
| System requirements analysis | Use test/performance records to confirm ability to meet needs | Test to confirm ability to meet needs | Use test/performance records to confirm ability to meet needs | Test to confirm ability to meet needs | Use tests or records to determine potential to meet needs |
| 5.4.1 System wide design | Consider software's capabilities and characteristics in designing system behavior and in making other system-wide design decisions | | | | |
| 5.4.2 System architectural design | Include the CSCI in the system architecture; allocate system requirements and system-wide design decisions to it | | Consider unit's capabilities and characteristics in designing CSCI's and allocating system requirements and system-wide design decisions to them | | |
| 5.5 Software requirements analysis | Specify the project-specific requirements the CSCI must meet; verify via records or retest that the CSCI can meet them | | Consider unit's capabilities and characteristics in specifying the requirements for the CSCI of which it is a part | | |
| 5.6.1 CSCI-wide design | No requirement: the CSCI-wide design decisions have already been made (recording the "as built" design is under 5.13) | | Consider unit's capabilities and characteristics in designing CSCI behavior and making other CSCI-wide design decisions. | | |
| 5.6.2 CSCI architectural design | No requirement: the CSCI's architecture is already defined (recording the "as built" design is under 5.13) | | Include the unit in the CSCI architecture and allocate CSCI requirements and CSCI-wide design decisions to it | | |
| 5.6.3 CSCI detailed design | No requirement: the CSCI's detailed design is already defined (recording the "as built" design is under 5.13) | | No requirement: the unit is already designed (recording the "as built" design is under 5.13) | | Modify the unit's design as needed |
| 5.7.1 Software Implementation | No requirement: the CSCI's units are already implemented | | No requirement: the unit is already implemented | | Modify the software for the unit's |
| 5.7.2-5.7.5 Unit testing | No Requirement: The CSCI's units have already been tested | Perform selectively if in question and units are accessible | No Requirement: The unit has already been tested | Perform this testing | |
| 5.8 Unit integration and testing | No Requirement: The CSCI's units have already been integrated | Perform selectively if in question and units are accessible | Perform except where integration is already tested/proven | Perform this testing | |
| 5.9 CSCI qualification testing | No Requirement: CSCI's is already tested and proven | Perform this testing | Include the unit in CSCI qualification testing | | |

Figure 9. Interpreting CG-SDDS for incorporation of reusable software

FIGURE 9 Interpreting CG-SDDS for incorporation of reusable software

| Interpret the activity as follows for each type of existing, reusable software | | | | | |
|---|--|--|---|--------------------------------------|---|
| If this CG SDDS activity is required: | For CSCI's to be used unmodified | | For software units to be used unmodified | | For software units being modified for/during project |
| | Positive Performance Record | No or Poor Performance Record | Positive performance record | No or Poor Performance Record | |
| 5.10 CSCI/HWCI integration and testing | Perform except where integration is already tested/proven | Include the CSCI in CSCI/ HWCI integration and testing | Include the unit in CSCI/HWCI integration and testing | | |
| 5.11 System qualification testing | Include the CSCI in system qualification testing | | Include the unit in system qualification testing | | |
| 5.12 Preparing for software use | Include the software for the CSCI or unit in the executable software; include in version descriptions; handle any license issues; cover use of the CSCI or unit, as appropriate, via existing, new or revised user/operator manuals; install the CSCI or unit as part of the overall system; include its use, as appropriate, in the training offered. | | | | |
| 5.13 Preparing for software transition | Include the software for the CSCI or unit in the executable software; prepare source files for the CSCI or unit, if available; include in version descriptions; handle any license issues; prepare or provide "as built" design descriptions for software whose design is known; install the CSCI or unit at the support site; demonstrate regenerability if source is available; include in the training offered. | | | | |
| 5.14 Software configuration management | Apply to all software products prepared, modified or used in incorporating this software | | | | |
| 5.15 Software product evaluation | Apply to all software products prepared, or modified in incorporating this software; for software products used unchanged, apply unless evidence of past evaluations indicates that such an evaluation would be duplicative | | | | |
| 5.16 Software quality assurance | Apply to all activities performed and all software products prepared, modified or used in incorporating this software | | | | |
| 5.17 Corrective action | Apply to all activities performed and all software products prepared, or modified in incorporating this software | | | | |
| 5.18 Joint reviews | Cover the software products prepared or modified in incorporating this software | | | | |
| 5.19 Other activities | Apply the full requirements of this section | | | | |

FIGURE 9 Interpreting CG-SDDS for incorporation of reusable software

Figure 9. Interpreting CG-SDDS for incorporation of reusable software

INDEX

Acronyms and abbreviations App.A
Analyzing test results 5.10.6, 5.11.6,
5.12.6, 5.13.6
Application of CG-SDDS 1.2, App.G,H,I
Associate contractors 5.20.5
Category classifications 5.18.2.c,
App.C
CDRL 1.2.2, 1.2.4, 3.14.4, 5.1.1, 6.2,
6.3, 6.4, 6.5, App.D.3.10, H
Coding and unit testing 5.8
Coding software units 5.8.1
Coding standards 5.2.5, App.B,D
Computer center software operator manuals
5.14.2
Computer instruction set architecture
manuals 5.14.5
Computer system operator manuals 5.14.4
Configuration control 5.17, 5.17.1,
5.17.1.2
Configuration identification 5.17.1.1
Configuration management 5.17
Configuration status accounting
5.17.2.2
Corrective action 5.18, 5.18.1, 5.18.2
CSCI architectural design 5.6.2
CSCI behavioral design 5.6.1
CSCI detailed design 5.7.1
CSCI engineering requirements 5.5.1
CSCI integration and test cases 5.11.1,
5.11.3
CSCI integration and test planning
5.1.2
CSCI integration and test procedures
5.11.2, 5.11.3
CSCI integration and test results
5.11.6
CSCI integration and testing 5.11,
5.11.4
CSCI interface design 5.7.2
CSCI interface requirements 5.5.2
CSCI test cases 5.10.1
CSCI test planning 5.1.2
CSCI test procedures 5.10.2, 5.10.3
CSCI test results 5.10.6
CSCI testing 5.10, 5.10.4
Data Item Descriptions (DIDs) 6.3,
App.H
Database design document 5.6.3, 5.7.3
Database logical design 5.6.3
Database physical design 5.7.3
Database population 5.8.2
Define (meaning in CG-SDDS) 1.2.6
Definitions (general) 3
Definitions (of evaluation criteria)
App.D
Deliverables 6.2, 6.3, 6.4, 6.5, App.H
Delivery (preparation for) 5.15
Design standards 5.2.5
Detailed requirements 5
Develop (meaning in CG-SDDS) 1.2.6
DIDs 6.3, App.H
Dry run of tests 5.10.3, 5.11.3,
5.12.3, 5.13.3
Engineering Change Proposals 5.17.2.1
Evaluation criteria App.D
Executable code (delivery of) 5.15.1,
6.2
Final software product evaluations
5.16.2
Functional Configuration Audit 5.15.5
General requirements 4
Government agencies (as developers)
1.2.3
Government Configuration Management
5.17.2
Independence in test/evaluation 4.3.1,
5.16.3
In-process evaluations 5.16.1
Installation at user sites 5.1.5,
5.14.7
Interface design document 5.7.2
Interface requirements specification
5.5.2
IV&V agents (interface with) 5.20.4
Joint reviews 5.19, App.E
Joint management reviews 5.19.2, App.E
Joint technical reviews 5.20.1
Life cycle models 4.1, 5, App.G
Management indicators 5.20.6, App.F
Management reviews (joint) 5.18.2,
App.E
Non-deliverable software 5.2.6
Notes (section) 6
Operational concept 5.3.2
Physical Configuration Audit 5.15.6
Populating databases 5.8.2
Preparation for software use and support
5.14
Priority classifications App.C
Privacy 5.20.2
Problem categories/priorities 5.18.2.c,
App.C
Problem/change reports 5.18.1, 5.18.2.a
Process improvement 5.18, 5.18.1,
5.18.2
Processing resource/reserve capacity
4.2.4
Project planning 5.1
Purpose of CG-SDDS 1.1
Rationale (recording) 4.2.5
Record (meaning in CG-SDDS) 1.2.7
Referenced documents 2
Reserve requirements 4.2.4
Reusable software 4.2.2, App.I
Revision and retesting 5.8.5, 5.9.3,
5.10.5,
5.11.5, 5.12.5, 5.13.5
Risk management 5.20.1
Safety analysis 4.2.3
Scope of CG-SDDS 1
Security 5.20.2
Software architectural design 5.6
Software configuration management 5.17
Software delivery (preparing for) 5.15
Software design document 5.4.1, 4.4.2
5.6.1, 5.6.2, 5.7.

Software detailed design 5.7, 5.7.1
Software development environment 5.2
Software development files 5.2.4, 5.8.3, 5.8.5, 5.8.6, 5.9.1, 5.9.3, 5.9.4, 5.10.3, 5.10.5, 5.11.3, 5.11.5, 5.12.3, 5.12.5, 5.13.3, 5.17.1
Software development library 5.2.3
Software development methods 4.2.1
Software development plan 5.1.1
Software development process 4.1, 5, App.G
Software engineering (general requirements) 4.2
Software engineering environment 5.2.1
Software input/output manuals 5.14.3
Software installation plan 5.1.5
Software management indicators 5.20.6, App.F
Software manuals 5.14
Software product evaluations 5.16, App.D
Software product evaluation records 5.16.4
Software product specification 5.15.3
Software requirements analysis 5.5
Software requirements specification 5.5.1
Software reviews 5.19, App.E
Software support plan 5.1.4
Software test description 5.10.1, 5.10.2, 5.11.1, 5.11.2, 5.12.1, 5.12.2, 5.13.1, 5.13.2
Software test environment 5.2.2
Software test plan 5.1.2, 5.1.3
Software test document 5.10.6, 5.11.6, 5.12.6,
Software testing 4.3, 5.8 - 5.13
Software user manuals 5.14.1
Source code (delivery of) 5.15.2, 5.13.6
Storage, handling, and delivery 5.17.3
Stress testing 4.3.3, 5.8.3, 5.9.1, D
Subcontractor management 5.20.3
System architectural design 5.4.2
System behavioral design 5.4.1
System design 5.4
System development 1.2.5
System requirements 5.3.3
System requirements analysis 5.3
System/segment specification 5.3.3
System test cases 5.12.1, 5.12.2, 5.12.4
System test planning 5.1.3
System test procedures 5.12.2, 5.12.3
System test results 5.12.6
System testing 5.12
Tailoring CG-SDDS 1.2.2, App.G
Technical reviews (joint) 5.18.1
Target computer system 4.3.2
Training 5.14.6.b, 5.14.7.c
Transition to support site 5.1.4, 5.14.7
Unit integration and testing 5.9, 5.9.1-4
Unit testing 5.8, 5.8.3, 5.8.4, 5.8.5, 5.8.6
User input, analyzing 5.3.1
Version description document 5.15.4

| | |
|---|--|
| COAST GUARD DATA ITEM DESCRIPTION | APPROVAL DATE: |
| Public reporting burden for collection of this information is estimated to average 110 hours per response, including the time for reviewing instructions, searching for existing data sources, gathering and maintaining the data needed and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to COMMANDANT (G-SCC-2), 2100 Second Street S.W., Room 6410, Washington, DC 20593-0001. | |
| 1. TITLE: Coast Guard Software Plan (CG-SP) | 2. IDENTIFICATION NUMBER: DI-CGSDDS-001 |
| 3. DESCRIPTION/PURPOSE: 3.1 The Coast Guard Software Plan (CG-SP) is a single document encompassing the planning for software development, software installation, and preparation for transition to software support. | |
| 4. CLIN NUMBER(s): | 5. OFFICE OF PRIMARY RESPONSIBILITY: USCG Software Policy & Standards Division (G-SCC-2) |
| 6. APPLICATION/INTERRELATIONSHIP: 6.1 This Data Item Description (DID) contains the format and contents preparation instructions for data generated under the work task described in paragraphs 5.1.1, 5.1.4, 5.1.5, 5.21.1, 5.21.2, and 5.21.3 of CG-SDDS (COMDTINST 5234.4). 6.2 Commercial, or other existing documents may be substituted for all or part of this document if they contain the required data. | |
| 7. PREPARATION INSTRUCTIONS: 7.1 <u>General Instructions.</u> The document shall be developed according to the following format specifications: a. <u>Automated Techniques.</u> Use of automated techniques is encouraged. The Contract Data Requirements List (CDRL) should specify whether the data are to be delivered on paper or electronic media; any requirements on the electronic representation (such as ASCII, CALS, or compatibility with a specific word processor or other software); whether the data may be delivered in contractor format rather than in the format specified herein; and whether the data may reside in a computer-aided software engineering (CASE) or other automated tool rather than in the form of a traditional document. b. <u>Alternate presentation styles.</u> Charts, tables, matrices, and other presentation styles are acceptable substitutes for text when data required by this DID can be made more readable using these styles. (Continued on Page 2) | |
| 8. DISTRIBUTION STATEMENT: <u>DISTRIBUTION STATEMENT A.</u> This DID will be incorporated into a COMDTINST along with the approved CG Software Development & Documentation Standards (CG-SDDS). | |

7. PREPARATION INSTRUCTIONS (continued)

- c. **Title page or identifier.** When data are delivered in the form of a traditional document, the document shall include a title page containing, as applicable: document number; volume number; version/revision indicator; security markings or other restrictions on the handling of the document; date; document title; name, abbreviation, and any other identifier for the system, subsystem or item to which the document applies; contract number; CDRL item number; organization for which the document has been prepared; name and address of preparing organization. For data delivered in an alternative form, this information shall be included on external and internal labels or by equivalent identification methods.
- d. **Table of contents.** When data are delivered in the form of a traditional document, the document shall contain a table of contents providing the number, title, and page number of each titled paragraph, figure, table, and appendix. For data delivered in an alternative form, this information shall consist of an internal or external table of contents containing pointers to, or instructions for accessing, each paragraph, figure, table, and appendix or their equivalents.
- e. **Page numbering/labeling.** When data are delivered in the form of a traditional document, each page shall contain a unique page number and display the document number, including version, volume, and date, as applicable. For data delivered in an alternative form, files, screens, or other entities shall be assigned names or numbers in such a way that desired data can be indexed and accessed.
- f. **Response to tailoring instructions.** When data are delivered in the form of a traditional document, paragraphs that have been tailored out of the DID shall result in the corresponding paragraph number and title in the document, followed by "This paragraph has been tailored out." For data delivered in an alternative form, this representation need occur only in the table of contents or equivalent.
- g. **Multiple paragraphs and subparagraphs.** Any section, paragraph, or subparagraph in this DID may be written as multiple paragraphs or subparagraphs to enhance

readability

7.2 Content. Content requirements begin on the following page. The numbers shown designate the paragraph numbers to be used in the document. Each such number is understood to have the prefix "7.2" within this DID. For example, the paragraph numbered 1.1 is understood to be paragraph 7.2.1.1 within this DID.

1. **Scope.**
 - 1.1 **Identification.** This paragraph shall contain a full identification of the system and the Computer Software Configuration Items (CSCIs) to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s).
 - 1.2 **System overview.** This paragraph shall briefly state the purpose of the system and the software to which this document applies. It shall describe the general nature of the system and software; summarize the history of system development, operation, and maintenance; identify the project sponsor, user, developer, and support agencies; identify current and planned operating sites; and list other relevant documents.
 - 1.3 **Document overview.** This paragraph shall summarize the purpose and contents of this document.
2. **Referenced documents.** This section shall list by document number and title all documents referenced in this document. This section shall also identify the source for all documents not available through normal Government stocking activities.
3. **Software development planning.** This section shall be divided into the following paragraphs to describe the contractor's plans for conducting the software development effort. (Note: When the software development is performed in-house, the term "contractor" will be understood to mean "in-house developer")
 - 3.1 **Overview of the work to be done.** This paragraph shall be divided into subparagraphs as needed to provide an overview of the work to be accomplished and to establish the context for the planning described in later sections. It shall include, as applicable, contractual and other requirements and constraints regarding:
 - a. The system and software to be developed
 - b. The documentation required
 - c. Overview of the system life cycle and the position of the project within that life cycle

- d. The software life cycle model to be used
- e. Project schedules and resources
- f. Other aspects of the project, such as security, privacy, methods, standards to be followed, testing constraints, etc.

3.2 Complying with Coast Guard Software Documentation Standards (CG-SDDS) general requirements.

This paragraph shall be divided into subparagraphs as needed to provide the following information. Provisions corresponding to non-required activities may be satisfied by the words "Not applicable." Each subparagraph shall identify applicable risks/uncertainties and plans for dealing with them.

a. The software development process to be used, including:

- 1) A proposed life cycle model for the software if none is established in the contract, that is, a set of phases, builds, increments, blocks, or other project stages, and the objectives of each. (Note: The remainder of this DID uses the term "build;" this term should be interpreted to suit the project)
- 2) A mapping of the activities required by CG-SDDS (and other contract provisions) onto that portion of the software life cycle model covered by the contract, that is, identification of the activities to be performed in each build and, as applicable, their relationships to one another

b. General plans for software engineering, including:

- 1) The software development methods to be used, including the tools and procedures to be used in support of these methods
- 2) The approach to be followed for identifying, evaluating, and incorporating reusable software, including identification of any candidate or selected reusable components known at the time this plan is prepared
- 3) The approach to be followed for safety analysis

- 4) The approach to be followed for analyzing, allocating, and monitoring the processing resource and reserve requirements established for the software
- 5) The approach to be followed for recording the rationale for key decisions

c. General plans for software testing, including the approach to be followed for:

- 1) Achieving the required level of independence
- 2) Including testing on the target computer system or an equivalent system
- 3) Stressing the software at the limits of its specified requirements

3.3 Complying with CG-SDDS detailed requirements. This paragraph shall be divided into the subparagraphs as needed to provide the following information. Provisions corresponding to non-required activities may be satisfied by the words "Not applicable." The discussion of each activity shall include the approach (methods, procedures, and/or tools) to be applied to 1) the analysis or other technical tasks involved, 2) the recording of results, and 3) the preparation of associated deliverables, if applicable. The discussion shall also identify applicable risks/uncertainties and plans for dealing with them.

a. The approach to be followed for developing project plans in each build, including:

- 1) Further development of this software development plan
- 2) Planning CSCI and CSCI integration testing
- 3) Performing or participating in planning system testing
- 4) Planning for transition to software support
- 5) Planning for software installation and training at user sites

b. The approach to be followed for establishing, controlling, and maintaining a software development environment in each build, including descriptions of:

- 1) The software engineering environment
- 2) The software test environment

- 3) The software development library
 - 4) The software development files
 - 5) Design and coding standards to be used
 - 6) Non-deliverable software to be used
 - 7) Any other software standards and procedures to be used
- c. The approach to be followed for performing or participating in system requirements analysis in each build, including:**
- 1) Analyzing user input
 - 2) Defining the operational concept
 - 3) Defining the system requirements
- d. The approach to be followed for performing or participating in system design analysis in each build, including:**
- 1) Developing the system behavioral design
 - 2) Developing the system architectural design
- e. The approach to be followed for software requirements analysis in each build, including:**
- 1) Defining the CSCI engineering (and corresponding qualification) requirements
 - 2) Defining the CSCI interface (and corresponding qualification) requirements
- f. The approach to be followed for performing software architectural design in each build, including:**
- 1) Developing the CSCI behavioral design
 - 2) Developing the CSCI architectural design
 - 3) Developing the database logical design
- g. The approach to be followed for performing software detailed design in each build, including:**
- 1) Developing the CSCI detailed design
 - 2) Developing the CSCI interface design
 - 3) Developing the database physical design
- h. The approach to be followed for coding and unit testing in each build, including the programming language(s) to be used, including:**

- 1) Coding software units
 - 2) Populating those databases to be populated as part of software development
 - 3) Preparing for unit testing
 - 4) Performing unit testing
 - 5) Revision and retesting based on test results
 - 6) Recording unit test results
- i. The approach to be followed for unit integration and testing in each build, including:**
- 1) Preparing for unit integration and testing
 - 2) Performing unit integration and testing
 - 3) Revision and retesting based on test results
 - 4) Recording unit integration and test results
- j. The approach to be followed for CSCI testing in each build, including:**
- 1) Preparing CSCI test cases
 - 2) Preparing CSCI test procedures
 - 3) Dry run of CSCI test procedures
 - 4) Performing CSCI testing (including Government witnessed testing, as applicable)
 - 5) Revision and retesting based on test results
 - 6) Analyzing and recording CSCI test results
 - 7) Updating CSCI test cases and procedures
- k. The approach to be followed for CSCI integration and testing in each build, including:**
- 1) Preparing CSCI integration and test cases
 - 2) Preparing CSCI integration and test procedures
 - 3) Dry run of CSCI integration and test procedures
 - 4) Performing CSCI integration and testing (including Government witnessed testing, as applicable)
 - 5) Revision and retesting based on test results
 - 6) Analyzing and recording CSCI integration and test results
 - 7) Updating CSCI integration and test cases and procedures
- l. The approach to be followed for performing or participating in system testing in each build, including:**
- 1) Preparing system test cases

- 2) Preparing system test procedures
 - 3) Dry run of system test procedures
 - 4) Performing system testing (including Government witnessed testing, as applicable)
 - 5) Revision and retesting based on test results
 - 6) Analyzing and recording system test results
 - 7) Updating system test cases and procedures
- m. The approach to be followed for preparing for software use and support in each build, including:**
- 1) Developing software user manuals
 - 2) Developing computer center software operator manuals
 - 3) Developing software input/output manuals
 - 4) Developing computer system operator manuals
 - 5) Performing installation and training at user sites
 - 6) Transitioning software and environments to the designated support site
- n. The approach to be followed for preparing for software delivery in each build, including:**
- 1) Preparing executable code for delivery
 - 2) Preparing source code for delivery
 - 3) Developing software product specifications
 - 4) Developing version descriptions
 - 5) Supporting Functional Configuration Audit(s)
 - 6) Supporting Physical Configuration Audit(s)
- o. The approach to be followed for performing software product evaluations of contractor and subcontractor products in each build, including a summary of the following topics:**
- 1) Performing inprocess software product evaluations
 - 2) Performing final software product evaluations
 - 3) Achieving the required independence in software product evaluation activities
 - 4) Preparing and maintaining software product evaluation records, including items to be recorded
- p. The approach to be followed for performing software configuration management in each build, including a summary of the following topics:**

- 1) Controlling development products, including:
 - a) Configuration identification of development products
 - b) Configuration control of development products
 - 2) Interface with Government Configuration Management, including:
 - a) Supporting the baselining of specifications
 - b) Using Engineering Change Proposals
 - c) Configuration status accounting, including the format, content, and purpose of reports to be used
 - 3) Storage, handling, and delivery of project media
- q. The approach to be followed for performing corrective action and process improvement in each build, including a summary of the following topics:**
- 1) Preparing problem/change reports, including the items to be included
 - 2) Implementing a corrective action and process improvement system
- r. The approach to be followed for holding joint (customer/contractor) reviews in each build, including:**
- 1) A proposed set of technical level reviews, including the items to be reviewed, objectives, and preparatory and followup activities for each review
 - 2) A proposed set of management level reviews, including the items to be reviewed, objectives, and preparatory and followup activities for each review
- s. The approach to be followed for performing software development management in each build, including a summary of the following topics:**
- 1) Risk management
 - 2) Complying with the security and privacy requirements in the contract
 - 3) Managing subcontractors
 - 4) Interfacing with software IV&V agents
 - 5) Coordinating software development efforts to ensure

compatibility at interfaces with associate contractors

- 6) Applying management indicators, including indicators to be used, data to be collected, methods for interpreting/applying the data, and reporting mechanisms

t. A summary of the approach to be followed for performing any other software development activities in each build.

3.4 Schedules. This paragraph shall present the schedule(s) for the project. It shall include:

- a. Schedule(s) identifying the activities in each build and showing initiation of each activity, availability of draft and final deliverables and other milestones, and completion of each activity
- b. An activity network, depicting sequential relationships and dependencies among activities and identifying those activities that impose the greatest time restrictions on the project

3.5 Project organization and resources. This paragraph shall be divided into subparagraphs as needed to describe:

- a. The organizational structure to be used on the project, including the organizations involved, their relationships to one another, and the authority and responsibility of each organization for carrying out required activities
- b. The resources to be applied to the project, including:
 - 1) Personnel resources:
 - a) Estimated staff-loading (number of personnel over time)
 - b) Breakdown of the staff-loading numbers by responsibility
 - c) Overview of the skill levels, geographic locations, and security clearances
 - 2) Overview of contractor facilities to be used
 - 3) Government furnished items facilities required and dates needed
 - 4) Other required resources, plan for obtaining them, and need/availability dates

4. **Software installation planning.** This section shall be divided into the following paragraphs to describe the contractor's plans for installing the software at user sites.
- 4.1 **Installation overview.** This paragraph shall be divided into subparagraphs as needed to provide the following information:
- a. A general description of the installation process, including list of sites and installation schedule
 - b. The organizational name, office symbol/code, and telephone number of a contact for questions relating to installation
 - c. A list of support materials required for the installation, including magnetic tapes, disk packs, computer printer paper, and special forms
 - d. A description of the briefings, seminars, and training to be provided
 - e. A list and brief description of each task required for the system installation, including responsible organization
 - f. A description of the number and skill level of the personnel required during the installation period and the days and times they will be needed, including the need for multi-shift operation, clerical support, etc.
 - g. An overview of security considerations associated with the system
- 4.2 **Site information for computer operations personnel.** This paragraph applies if the system will be installed in computer center(s) for users to access via terminals or using batch inputs/ outputs. If this type of installation does not apply, this paragraph shall contain the words "Not applicable."
- 4.2.1 **(Site Name).** This paragraph shall identify a site or set of sites and provide the following information for those sites.

- a. Schedule of activities to be accomplished during installation
- b. Inventory of software required to support the installation, including name, identification code or acronym, security classification, and whether the software is expected to be on site or will be delivered for the installation
- c. Physical facilities and accommodations required during the installation period, including classroom, work space, training aids, hardware, transportation, and lodging, and the days and times they will be needed
- d. Composition of the installation team, and each member's tasks
- e. Step-by-step procedures for accomplishing the installation and conversion from the old system
- f. Data update procedures during the installation period, if different from normal, including step-by-step procedures for updating the converted data

4.3. Site information for user personnel. This section shall provide users with the information necessary to accomplish an orderly installation. If more than one type of user is involved, separate information shall be provided for each, as applicable.

4.3.1 (Site Name). This paragraphs shall identify a site or set of sites and shall provide the following information for those sites:

- a. Schedule of activities to be accomplished by the user during installation
- b. Step-by-step procedures for accomplishing the installation and conversion
- c. The user's data update procedures during the installation period, if different from normal, including step-by-step procedures for updating the converted data

5. Software support planning. This section shall be divided

into the following paragraphs to describe the contractor's plans for transitioning the software to the support agency.

5.1 Software support resources. This paragraph shall be divided into subparagraphs as needed to identify and describe the components of the software engineering and test environments required to support the deliverable software (including modify, document, test, copy, distribute, and control). Items to be described shall include the following, as applicable:

- a. Facilities: buildings or building features required for support; mock-ups; power requirements; security and safety measures
- b. Hardware: computer and other equipment required for support, including specific model numbers, acceptable alternatives, rationale for including, status of each item (Government-furnished, deliverable, already owned by support agency, must be acquired), information about where to acquire, known limitations, reference to user/operator manuals or instructions
- c. Software: software and associated documentation require for support, including titles, version numbers, acceptable alternatives, rationale for including, status of each item (Government-furnished, deliverable, already owned by support agency, must be acquired), information about where to acquire, known limitations, reference to user/operator manuals or instructions
- d. Other documentation needed for support, including titles, numbers, versions; acceptable alternatives, rationale for including, status of each document (Government-furnished, deliverable, already owned by support agency, must be acquired), information about where to acquire, known limitations
- e. Personnel needed to support the deliverable software, including number of personnel, types and levels of skills and expertise, and security clearances. This paragraph shall cite, as applicable, actual staffing on the development project, as applicable to estimate how to staff the support effort

- f. Other resources required for support, including consumables such as magnetic tapes, together with an estimate of the type and number that should be acquired
- g. The interrelationship of the components identified in the preceding paragraphs
- h. Data: identify and describe the data and associated documentation required to support the deliverable software. This data may include CASE tool repositories, test data, configuration and other databases/data banks, and other data. The description shall include:
 - 1) Specific names, identification numbers, version numbers, release numbers and configurations, as applicable, and acceptable alternatives
 - 2) Rationale for the selected data
 - 3) A figure showing the interrelationship of the data
 - 4) Reference to user/operator manuals or instructions for each item, as applicable
 - 5) Identification of each identified item and document as Government furnished, an item that will be delivered to the support agency, an item the support agency is known to have, an item the support agency must acquire, or other description of status
 - 6) If items must be acquired, information about where to acquire them security and privacy considerations, limitations, or other items of interest

5.2 Recommended procedures. This paragraph shall be describe procedures that the contractor may wish to recommend to the support agency for supporting the deliverable software. Included may be procedures for software support management, software engineering, software testing, software product evaluations, software configuration management, or other activities.

5.3 Training. This paragraph shall describe the contractor's plans for training personnel to manage and implement support of the deliverable software. The schedule and location for all required training shall be provided, as well as the delineation between classroom training and "hands-on" training. This paragraph shall provide (either directly or by reference) provisions for familiarization with the

operational software and target computer(s), the support software and host system, and equipment maintenance procedures, as applicable.

5.4 Anticipated areas of change. This paragraph shall describe the anticipated areas of change to the deliverable software.

5.5 Transition planning. This paragraph shall be divided into subparagraphs as needed to describe the contractor's plans for transitioning the deliverable software to the support agency. Included shall be descriptions of:

- a. All activities to be performed to transition the deliverable software to support
- b. Roles and responsibilities for each activity
- c. The resources required and the source of each resource
- d. Schedules and milestones for transition activities
- e. Procedures for installation and checkout of the deliverable software in the support environment designated by the contracting agency

5.6 Software quality assurance. This paragraph shall be divided into the following subparagraphs to describe the approach to be followed for software quality assurance. The planning in each subparagraph shall cover all requirements described in paragraphs 5.21, 5.21.1, 5.21.2, and 5.21.3 of the CG-SDDS.

6. Notes. This section shall contain any general information that aids in understanding this document (e.g., background information, glossary). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.

A. Appendixes. Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B,

etc.).

| | |
|---|--|
| COAST GUARD DATA ITEM DESCRIPTION | APPROVAL DATE: |
| Public reporting burden for collection of this information is estimated to average 110 hours per response, including the time for reviewing instructions, searching for existing data sources, gathering and maintaining the data needed and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to COMMANDANT (G-SCC-2), 2100 Second Street S.W., Room 6410, Washington, DC 20593-0001. | |
| 1. TITLE: Coast Guard Software Requirements Document (CG-SRD) | 2. IDENTIFICATION NUMBER: DI-CGSDDS-002 |
| 3. DESCRIPTION/PURPOSE: 3.1 The Software Requirements Document (CG-SRD) is a single document encompassing the operational concept for a system and the requirements for the system, for one or more Computer Software Configurations Items (CSCIs), and for the external interfaces of those CSCIs | |
| 4. CLIN NUMBER(s): | 5. OFFICE OF PRIMARY RESPONSIBILITY: USCG Software Policy & Standards Division (G-SCC-2) |
| 6. APPLICATION/INTERRELATIONSHIP: 6.1 This Data Item Description (DID) contains the format and contents preparation instructions for data generated under the work task described in paragraphs 5.3.2, 5.3.3, 5.5.1, and 5.5.2 of CG-SDDS (COMDTINST 5234.4). 6.2 Commercial, or other existing documents may be substituted for all or part of this document if they contain the required data. | |
| 7. PREPARATION INSTRUCTIONS: 7.1 <u>General Instructions.</u> The document shall be developed according to the following format specifications: <ul style="list-style-type: none"> a. <u>Automated Techniques.</u> Use of automated techniques is encouraged. The Contract Data Requirements List (CDRL) should specify whether the data are to be delivered on paper or electronic media; any requirements on the electronic representation (such as ASCII, CALS, or compatibility with a specific word processor or other software); whether the data may be delivered in contractor format rather than in the format specified herein; and whether the data may reside in a computer-aided software engineering (CASE) or other automated tool rather than in the form of a traditional document. b. <u>Alternate presentation styles.</u> Charts, tables, matrices, and other presentation styles are acceptable substitutes for text when data required by this DID can be made more readable using these styles. <p style="text-align: right;">(Continued on Page 2)</p> | |
| 8. DISTRIBUTION STATEMENT: DISTRIBUTION STATEMENT A. This DID will be incorporated into a COMDTINST along with the approved CG Software Development & Documentation Standards (CG-SDDS). | |

7. PREPARATION INSTRUCTIONS (continued)

- c. **Title page or identifier.** When data are delivered in the form of a traditional document, the document shall include a title page containing, as applicable: document number; volume number; version/revision indicator; security markings or other restrictions on the handling of the document; date; document title; name, abbreviation, and any other identifier for the system, subsystem or item to which the document applies; contract number; CDRL item number; organization for which the document has been prepared; name and address of preparing organization. For data delivered in an alternative form, this information shall be included on external and internal labels or by equivalent identification methods.
- d. **Table of contents.** When data are delivered in the form of a traditional document, the document shall contain a table of contents providing the number, title, and page number of each titled paragraph, figure, table, and appendix. For data delivered in an alternative form, this information shall consist of an internal or external table of contents containing pointers to, or instructions for accessing, each paragraph, figure, table, and appendix or their equivalents.
- e. **Page numbering/labeling.** When data are delivered in the form of a traditional document, each page shall contain a unique page number and display the document number, including version, volume, and date, as applicable. For data delivered in an alternative form, files, screens, or other entities shall be assigned names or numbers in such a way that desired data can be indexed and accessed.
- f. **Response to tailoring instructions.** When data are delivered in the form of a traditional document, paragraphs that have been tailored out of the DID shall result in the corresponding paragraph number and title in the document, followed by "This paragraph has been tailored out." For data delivered in an alternative form, this representation need occur only in the table of contents or equivalent.
- g. **Multiple paragraphs and subparagraphs.** Any section, paragraph, or subparagraph in this DID may be written as multiple paragraphs or subparagraphs to enhance readability

7.2 **Content.** Content requirements begin on the following page.

The numbers shown designate the paragraph numbers to be used in the document. Each such number is understood to have the prefix "7.2" within this DID. For example, the paragraph numbered 1.1 is understood to be paragraph 7.2.1.1 within this DID.

1. **Scope.** This section shall be divided into the following paragraphs.
 - 1.1 **Identification.** This paragraph shall contain a full identification of the system and the CSCIs to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s).
 - 1.2 **System overview.** This paragraph shall briefly state the purpose of the system and the software to which this document applies. It shall describe the general nature of the system and software; summarize the history of system development, operation, and maintenance; identify the project sponsor, user, developer, and support agencies; identify current and planned operating sites; and list other relevant documents.
 - 1.3 **Document overview.** This paragraph shall summarize the purpose and contents of this document.
2. **Referenced documents.** This section shall list by document number and title all documents referenced in this document. This section shall also identify the source for all documents not available through normal Government stocking activities.
3. **Operational concept.** This section shall be divided into the following paragraphs to describe a proposed system in terms of the user needs it will fulfill, its relationship to existing systems or procedures, and the ways it will be used.
 - 3.1 **Current system or situation.** This paragraph shall be divided into subparagraphs as needed to describe:
 - a. The background, objectives, and scope of the current system or situation
 - b. Any operational policies and constraints that apply to the current system or situation
 - c. The current system or situation, including variations in

different states and modes of operation (e.g., regular, maintenance, training, degraded, emergency, wartime):

- 1) The operational environment and its characteristic
 - 2) Major components and the interconnections among these components
 - 3) Interfaces to external systems or procedures
 - 4) Capabilities/functions of the current system
 - 5) Chart and description of inputs, outputs, data flow, and manual/automated processes
 - 6) Performance characteristics, such as speed, throughput, volume, frequency
 - 7) Quality attributes, such as reliability, maintainability, availability
 - 8) Provisions for safety, security, privacy, and continuity of operations
- d. The types of users of the system, or personnel involved in the current situation, including, as applicable, organizational structures, training/skills, responsibilities, and interactions with one another
- e. The support concept and environment for the current system

3.2 Justification for and nature of changes. This paragraph shall be divided into subparagraphs as needed to describe:

- a. The justification for change, including:
- 1) New or modified aspects of user needs, missions, objectives, environments, interfaces, personnel or other factors that require a new or modified system
 - 2) Deficiencies or limitations in the current system or situation that make it unable to respond to these factors
- b. New or modified capabilities or other changes needed to respond to these factors
- c. Priorities among the changes
- d. Changes considered but not included
- e. Assumptions and constraints applicable to the changes

3.3 Concept for a new or modified system. This paragraph shall be divided into subparagraphs as needed to describe a new or modified system. It shall describe, as applicable:

- a. The background, objectives, and scope of the new or modified system
- b. Any operational policies and constraints that apply to the new or modified system
- c. The new or modified system, including variations in different states and modes of operation (e.g., regular, maintenance, training, degraded, emergency, wartime):
 - 1) The operational environment and its characteristics
 - 2) Major components and the interconnections among these components
 - 3) Interfaces to external systems or procedures
 - 4) Capabilities/functions of the new or modified system
 - 5) Chart and description of inputs, outputs, data flow, and manual/automated processes
 - 6) Performance characteristics, such as speed, throughput, volume, frequency
 - 7) Quality attributes, such as reliability, maintainability, availability
 - 8) Provisions for safety, security, privacy, and continuity of operations
- d. The types of users of the system, including, as applicable, organizational structures, training/skills, responsibilities, and interactions with one another
- e. The support concept and environment for the new or modified system

3.4 Operational scenarios. This paragraph shall describe one or more operational scenarios that illustrate the role of the new or modified system, its interface to other systems, and all states or modes identified for the system.

3.5 Summary of impacts. This paragraph shall be divided into subparagraphs as needed to describe the operational and organization impacts on the user, development, and support/maintenance agency(ies), and the anticipated impacts on

these organizations during the development process.

- 3.6 Analysis of the proposed system.** This paragraph shall be divided into subparagraphs as needed to provide:
- a. A qualitative and quantitative summary of the benefits to be obtained from the new or modified system, including new and enhanced capabilities or performance
 - b. A qualitative and quantitative summary of disadvantages or limitations of the new or modified system
 - c. A description of major alternatives considered, the tradeoffs among them, and rationale for the decisions reached
- 4. System/segment specification.** This section shall be divided into the following paragraphs to specify the requirements (conditions for acceptance) for the system or segment. Upon Government approval and authentication, this section becomes the Functional Baseline for the system or segment. The word "system" may be interpreted to mean "segment" as applicable in this section.

4.1 Requirements.

- 4.1.1 Definition.** This paragraph shall provide a brief description of the system, identifying any states or modes in which the system is required to operate (for example, idle, ready, active, post use analysis, training, degraded, emergency, backup, wartime, peacetime). Note: the distinction between states and modes is arbitrary. A system may be described in terms of states only, modes only, states within modes, modes within states, or any other scheme that seems useful. If the system operates without states or modes, this paragraph shall so state, without the need to create artificial distinctions.
- 4.1.2 Characteristics.** This paragraph shall be divided into the following subparagraphs to describe the requirements for the system. Each requirement or group of requirements shall be annotated to indicate the states and modes in which it applies, either in the text, in an accompanying table, or by other means.

- 4.1.2.1 Performance characteristics.** This paragraph shall be divided into subparagraphs, each identifying a capability of the system by name and project unique identifier, describing its purpose, and itemizing the requirements for that capability. A "capability" is defined as a group of related requirements. The word "capability" may be replaced by "function," "subject," "object," or other term useful for presenting the requirements. Each requirement shall be given a unique identifier. The requirements shall include applicable parameters, such as response times, sequencing, accuracy, capacities (how much/how many), priorities, continuous operation requirements, and allowable deviations based on operating conditions, and shall express these parameters in measurable terms. The requirements shall also include required behavior under unexpected or "out of bounds" conditions and for maintaining continuity of operations in the event of emergencies.
- 4.1.2.2 Database/data bank requirements.** This paragraph shall be divided into subparagraphs as needed to identify each required database or data bank, state its purpose, and itemize the requirements imposed on it, including, as applicable, required data elements, required characteristics of the data elements, required relationships among data elements, and required storage capacity for the data, including growth capability.
- 4.1.2.3 External interface requirements.** This paragraph shall be divided into subparagraphs as needed to identify each external item with which the system is required to interface, state the purpose of the interface, describe the relationship between the interface and the states and modes of the system, and itemize the requirements imposed on the system as a result of the interface, including, as applicable, physical interface requirements (dimensions, tolerances, loads, etc.); communication/data transfer requirements; characteristics of the inputs that must be accepted by the system from the item; characteristics of the outputs that must be provided by the system to the item; security or privacy requirements.
- 4.1.2.4 Physical characteristics.** This paragraph shall specify any requirements for physical characteristics (e.g., weight limits, dimensional limits, color, protective coatings) of the system. Considerations for determining physical

requirements include transportation and storage, security, privacy, durability (freedom from corrosion, abrasion, or other damage), safety, vulnerability. If there are no physical requirements (such as for a software only system), this paragraph may be satisfied by "Not Applicable."

4.1.2.5 System quality factors. This paragraph shall be divided into subparagraphs as needed to specify any requirements pertaining to system quality factors, including, as applicable:

- a. Reliability (including failure contingencies)
- b. Maintainability
- c. Availability
- d. Other quality factors

4.1.2.6 Environmental requirements. This paragraph shall be divided into subparagraphs as needed to specify any requirements regarding the environment within which the system is required to operate, including, as applicable:

- a. Environmental conditions the system must withstand
- b. Computer equipment that must be used by, or incorporated into, the system
- c. Support software that must be used by, or incorporated into, the system
- d. The communications environment within which the system must operate

4.1.3 Design and construction. This paragraph shall be divided into subparagraphs as needed to specify any minimum system design and construction standards for the system requirements for system security and privacy, including required safeguards to reduce vulnerabilities.

4.1.4 Documentation. This paragraph shall any specify requirements for system documentation such as specifications, drawings, technical manuals, test plans and procedures, and installation instruction data.

4.1.5 Logistics. This paragraph shall specify any logistic considerations and conditions that apply to the operational requirements. These considerations and conditions may include maintenance, transportation modes, supply system requirements, impact on existing facilities, and impact on

existing equipment,

- 4.1.6 Personnel and training.** This paragraph shall be divided into subparagraphs as needed to specify:
- a. Personnel requirements that must be integrated into system design
 - b. Requirements for the training to be provided: responsibilities; required equipment; training devices to be developed; training times and locations; training materials
- 4.1.7 Characteristics of subordinate elements.** This paragraph shall be divided into subparagraphs as needed to identify and describe each segment of the system. This paragraph shall describe the relationships between the segments.
- 4.1.8 Precedence.** This paragraph shall specify any order of precedence or assigned weights indicating the relative importance of the requirements.
- 4.2 Quality assurance provisions.** This paragraph shall be divided into subparagraphs as needed to specify how compliance with the requirements of sections 4.1 and 4.3 is to be assured. It shall include:
- a. Definition of the qualification methods to be used (for example, demonstration, test, analysis, inspection)
 - b. Assignment of one or more of these methods to each requirement.
 - c. Description of any special tests or examinations to be performed, including requirements, as applicable, for standard samples, pre-production or periodic production samples, a pilot model, or a pilot lot.
- 4.3 Preparation for delivery.** This paragraph shall specify requirements for the preparation of the system and all its components for delivery, including packaging and handling.
- 5. Software requirements specification.** This section shall be divided into the following paragraphs to specify the engineering and qualification requirements for one or more Computer Software Configuration Item (CSCIs). Upon

Government approval and authentication, each section 5.x becomes the Allocated Baseline for the corresponding CSCI.

5.1 (CSCI name/identifier).

5.1.1 Engineering requirements. This paragraph shall be divided into the following subparagraphs to specify the engineering requirements (conditions for acceptance) for the CSCI. If the CSCI is part of a larger system, requirements to be included shall be the software requirements generated to satisfy the system or segment requirements allocated to this CSCI.

5.1.1.1 CSCI external interface requirements. This paragraph shall identify each required CSCIexternal interface and shall reference Section 6 or other document for the requirements for each interface.

5.1.1.2 CSCI capability requirements. This paragraph shall be divided into subparagraphs to identify each CSCI capability, state its purpose, and itemize the requirements associated with the capability. Capabilities may be divided into subcapabilities as needed, and the word "capability" may be replaced with "function," "subject," "object," or other term useful for presenting the requirements. The requirements shall include applicable parameters, such as response times, sequencing, accuracy, capacities (how much/how many), priorities, continuous operation requirements, and allowable deviations based on operating conditions, and shall express these parameters in measurable terms. The requirements shall also include required behavior under unexpected or "out of bounds" conditions and any provisions to be incorporated into the CSCI to provide continuity of operations in the event of emergencies. Each requirement shall be stated in such a way that an objective test can be defined for it. If the system of which the CSCI is a part can exist in various system states and modes, each CSCI requirement or group of requirements shall be correlated to those states and modes.

5.1.1.3 CSCI internal interfaces. This paragraph shall identify any requirements imposed on the interfaces between the capabilities identified above. If all internal interfaces are left to the design, this fact shall be so stated. Each internal interface on which requirements are imposed shall

be identified by name and project unique identifier and the interface requirements shall be itemized. Internal interface diagrams depicting data flow, control flow, and other relevant information may be used to aid in this description.

5.1.1.4 CSCI internal data element requirements. This paragraph shall specify any requirements imposed on the data elements internal to the CSCI. If all decisions about internal data elements are left to the design, this fact shall be so stated. If they have been covered elsewhere in this specification, they need not be repeated here. If requirements are imposed on data elements internal to the CSCI they shall include, as applicable:

- a. A project unique identifier for the data element
- b. A brief description of the data element
- c. Units of measure required for the data element, such as seconds, meters, kilohertz
- d. Limit/range of values required for the data element (for constants, the actual value)
- e. Accuracy required for the data element
- f. Precision or resolution required for the data element in terms of significant digits
- g. Required format, such as field definitions

5.1.1.5 Adaptation requirements. This paragraph shall be divided into subparagraphs as needed to specify any adaptation requirements for the CSCI, including, as applicable:

- a. Any site unique data required by each installation
- b. Any parameters required by the CSCI that may vary according to operational needs

5.1.1.6 Sizing and timing requirements. This paragraph shall specify:

- a. Sizing requirements on the CSCI, including, as applicable, amount/type/location of internal and auxiliary memory allocated to the CSCI; variations between normal operation and contingency operations; and other constraints imposed by the planned memory available to the CSCI
- b. Timing requirements on the CSCI, including, as

applicable, amount of processing time allocated to the CSCI; required throughput time; required response time to queries and other requests; sequential relationships of CSCIs; sequential relationships of CSCI capabilities; priorities imposed by type of inputs and modes of operation; timing requirements for range of loads under varying operating conditions; and required input/output transfer times

- 5.1.1.7 Safety requirements.** This paragraph shall specify any safety requirements applicable to the CSCI, concerning potential hazards to personnel, property, and the physical environment.
- 5.1.1.8 Security and privacy requirements.** This paragraph shall specify any security and privacy requirements applicable to the CSCI. These requirements shall include, as applicable:
- a. The type and degree of security or privacy of data to be used in or processed by the CSCI, including when the data become sensitive or change sensitivity
 - b. The type and degree of security or privacy of the algorithms that are permitted or required to be used in the CSCI
 - c. The security and privacy environment that can be assumed to exist during operation
- 5.1.1.9 Design constraints.** This paragraph shall specify any other requirements that constrain the CSCI design, such as the use of a particular processing configuration. If there are no constraints imposed on the CSCI design, this fact shall be so stated.
- 5.1.1.10 Software quality factors.** This paragraph shall be divided into subparagraphs as needed to specify any software quality factors identified in the contract or derived from a higher level specification and the method to be used to determine whether each quality factor has been met. These factors may include reliability (the ability to perform with correct, consistent results), maintainability (the ability to be easily corrected), availability (the ability to be accessed when needed), flexibility (the ability to be easily adapted to changing requirements), portability (the

ability to be easily modified for a new hardware environment), reusability (the ability to be used in multiple applications), testability (the ability to be easily and thoroughly tested), usability (the ability to be easily learned and used), and other attributes.

5.1.1.11 Human performance/human engineering requirements. This paragraph shall specify any human factors engineering requirements imposed on the CSCI. These requirements shall include, as applicable, considerations for:

- a. Human information processing capabilities and limitations
- b. Foreseeable human errors under both normal and extreme conditions
- c. Implications for the total system environment (include training, support, and operational environment)

5.1.1.12 Requirements traceability. This paragraph shall contain a mapping of the engineering requirements in this specification to the system requirements allocated to this CSCI, and a mapping of the system requirements allocated to this CSCI to the engineering requirements in this specification.

5.1.2 Qualification requirements. This paragraph shall specify the qualification requirements necessary to establish that each requirement in 5.x.1 and 5.x.3 has been met. It shall include, as applicable:

- a. Definition of the qualification methods to be used (for example, demonstration, test, analysis, inspection)
- b. Assignment of one or more of these methods to each requirement
- c. Description of any special tests or examinations to be performed

5.1.3 Preparation for delivery. This paragraph shall specify any requirements for delivery, including delivery media, labeling, packaging, handling, and classification markings.

6. Interface requirements specification. This section shall be divided into the following paragraphs to specify the requirements for one or more interfaces between one or more CSCIs and other configuration items or critical items.

These requirements may describe the interface characteristics of existing items or specify the required interface characteristics of new or to be modified items.

6.1 Interface requirements.

6.1.1 Interface identification and diagrams. This paragraph shall identify the interfaces to which this specification applies. These items may include systems, other configuration items, and support software and hardware, such as utilities and test software, databases, and items in the communications environment. The identification of each item shall include its name, number, version, and documentation references. The identification shall state which items already exist (and therefore impose interface requirements on interfacing items) and which are being developed or modified (thus having interface requirements imposed on them). One or more interface diagrams, as appropriate, shall be provided to depict the interfaces. Each interface shall be identified by name and project unique identifier, and shall specify, as applicable, the type of interface required (sequential or concurrent operation, real time data transfer, store and retrieve data transfer, operator controlled, etc.).

6.1.2 (Interface name and projectunique identifier). This paragraph (beginning with 6.1.2) shall identify an interface by name and projectunique identifier, shall state its purpose, and shall be divided into the following subparagraphs. When describing interface characteristics of existing items, read "established" for "required."

6.1.2.1 Data element requirements. This paragraph shall specify any requirements pertaining to data elements to be transmitted between the interfacing items, including, as applicable:

- a. A project-unique identifier for the data element
- b. A brief description of the data element
- c. The CSCI or other item that is the source of the data element, and an indication of which of these, if any, is imposing the requirement
- d. The CSCI(s) or other item(s) that are the recipients of the data element, and an indication of which of these, if any, is imposing the requirement
- e. The units of measure in which the data element must be

sent or received

- f. The limit/range of values that must be sent or received for the data element (for constants, the actual value; when applicable, allowable codes, message types)
 - g. The accuracy that must be possessed by the data element
 - h. The precision or resolution with which the data element must be sent or received, in terms of number of significant digits
 - i. The timing characteristics with which the data element must be sent or received (how often transmitted or received, transmitted for how long, etc.)
 - j. Legality checks the data element must be able to pass
 - k. The data type (such as integer, ASCII, real, enumerated, etc.) in which the data element must be sent or received
 - l. The data representation/format in which the data elements must be sent or received
 - m. Sequence or other dependencies among the data elements
- 6.1.2.2 Messages or other data assemblies. This paragraph shall specify any requirements concerning messages or other assemblies of data elements to be transmitted between the interfacing items. It shall identify each such message or assembly by name and projectunique identifier and shall describe the assignment of data elements to each message or assembly.

6.1.2.2 Messages or other data assemblies. This paragraph shall specify any requirements concerning messages or other assemblies of data elements to be transmitted between the interfacing items. It shall identify each such message or assembly by name and projectunique identifier and shall describe the assignment of data elements to each message or assembly.

6.1.2.3 Interface priorities. This paragraph shall specify any requirements concerning the relative priority of the interface, data elements, messages, or assemblies transmitted between the interfacing items.

6.1.2.4 Interface/communication protocols. This paragraph shall be divided into subparagraphs as needed to specify any requirements concerning commercial, military, or proprietary communications protocols to be used for the interface. It shall identify each protocol to be used and shall specify for each, as applicable, requirements for:

- a. Fragmentation and reassembly of messages
- b. Message formatting
- c. Legality checks, error control, and recovery procedures, including fault tolerance, handling of "outofbounds" conditions, and continuity of operations in emergencies
- d. Synchronization: connection establishment, maintenance, termination, timing

- e. Flow control, including sequence numbering, window size, and buffer allocation
- f. Data transfer rate, periodic or aperiodic, and minimum interval between transfers
- g. Routing, addressing, and naming conventions
- h. Transmission services, including priority and grade
- i. Status, identification, notification, and any other reporting features
- j. Security and privacy, including encryption, user authentication, compartmentalization, and auditing

6.2 Qualification requirements. This paragraph shall be divided into subparagraphs as needed to specify the qualification requirements necessary to establish that each requirement in 6.1 has been met. It shall include, as applicable:

- a. Definition of the qualification methods to be used (for example, demonstration, test, analysis, inspection)
- b. Assignment of one or more of these methods to each requirement
- c. Description of any special tests or examinations to be performed

7. Notes. This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, rationale). This section shall contain an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document, and a list of any terms and definitions needed to understand this document.

A. Appendixes. Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, etc.).

| | | |
|---|--|--|
| COAST GUARD DATA ITEM DESCRIPTION | | APPROVAL DATE: |
| Public reporting burden for collection of this information is estimated to average 110 hours per response, including the time for reviewing instructions, searching for existing data sources, gathering and maintaining the data needed and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to COMMANDANT (G-SCC-2), 2100 Second Street S.W., Room 6410, Washington, DC 20593-0001. | | |
| 1. TITLE: Coast Guard Software Requirements Document (CG-SDD) | | 2. IDENTIFICATION NUMBER: DI-CGSDDS-003 |
| 3. DESCRIPTION/PURPOSE: 3.1 The Software Design Document (CG-SDD) is a single document encompassing the design of a system, the design of each Computer Software Configuration Item (CSCI) in the system, the design of the CSCI-external interfaces, and the design of each database in the system. | | |
| 4. CLIN NUMBER(s): | | 5. OFFICE OF PRIMARY RESPONSIBILITY: USCG Software Policy & Standards Division (G-SCC-2) |
| 6. APPLICATION/INTERRELATIONSHIP: 6.1 This Data Item Description (DID) contains the format and contents preparation instructions for data generated under the work task described in paragraphs 5.4.1, 5.4.2, 5.6.1, 5.6.2, 5.6.3, 5.7.1, 5.7.2, and 5.7.3 of CG-SDDS (COMDTINST 5234.4). 6.2 Commercial, or other existing documents may be substituted for all or part of this document if they contain the required data. | | |
| 7. PREPARATION INSTRUCTIONS: 7.1 <u>General Instructions.</u> The document shall be developed according to the following format specifications: a. <u>Automated Techniques.</u> Use of automated techniques is encouraged. The Contract Data Requirements List (CDRL) should specify whether the data are to be delivered on paper or electronic media; any requirements on the electronic representation (such as ASCII, CALS, or compatibility with a specific word processor or other software); whether the data may be delivered in contractor format rather than in the format specified herein; and whether the data may reside in a computer-aided software engineering (CASE) or other automated tool rather than in the form of a traditional document. b. <u>Alternate presentation styles.</u> Charts, tables, matrices, and other presentation styles are acceptable substitutes for text when data required by this DID can be made more readable using these styles. (Continued on Page 2) | | |
| 8. DISTRIBUTION STATEMENT: DISTRIBUTION STATEMENT A. This DID will be incorporated into a COMDTINST along with the approved CG Software Development & Documentation Standards (CG-SDDS). | | |

7. PREPARATION INSTRUCTIONS (continued)

- c. **Title page or identifier.** When data are delivered in the form of a traditional document, the document shall include a title page containing, as applicable: document number; volume number; version/revision indicator; security markings or other restrictions on the handling of the document; date; document title; name, abbreviation, and any other identifier for the system, subsystem or item to which the document applies; contract number; CDRL item number; organization for which the document has been prepared; name and address of preparing organization. For data delivered in an alternative form, this information shall be included on external and internal labels or by equivalent identification methods.
- d. **Table of contents.** When data are delivered in the form of a traditional document, the document shall contain a table of contents providing the number, title, and page number of each titled paragraph, figure, table, and appendix. For data delivered in an alternative form, this information shall consist of an internal or external table of contents containing pointers to, or instructions for accessing, each paragraph, figure, table, and appendix or their equivalents.
- e. **Page numbering/labeling.** When data are delivered in the form of a traditional document, each page shall contain a unique page number and display the document number, including version, volume, and date, as applicable. For data delivered in an alternative form, files, screens, or other entities shall be assigned names or numbers in such a way that desired data can be indexed and accessed.
- f. **Response to tailoring instructions.** When data are delivered in the form of a traditional document, paragraphs that have been tailored out of the DID shall result in the corresponding paragraph number and title in the document, followed by "This paragraph has been tailored out." For data delivered in an alternative form, this representation need occur only in the table of contents or equivalent.
- g. **Multiple paragraphs and subparagraphs.** Any section, paragraph, or subparagraph in this DID may be written as multiple paragraphs or subparagraphs to enhance readability

7.2 **Content**. Content requirements begin on the following page. The numbers shown designate the paragraph numbers to be used in the document. Each such number is understood to have the prefix "7.2" within this DID. For example, the paragraph numbered 1.1 is understood to be paragraph 7.2.1.1 within this DID.

1. **Scope.** This section shall be divided into the following paragraphs.
 - 1.1 **Identification.** This paragraph shall contain a full identification of the system and the CSCIs to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s).
 - 1.2 **System overview.** This paragraph shall briefly state the purpose of the system and the software to which this document applies. It shall describe the general nature of the system and software; summarize the history of system development, operation, and maintenance; identify the project sponsor, user, developer, and support agencies; identify current and planned operating sites; and list other relevant documents.
 - 1.3 **Document overview.** This paragraph shall summarize the purpose and contents of this document.
2. **Referenced documents.** This section shall list by document number and title all documents referenced in this document. This section shall also identify the source for all documents not available through normal Government stocking activities.
3. **System/segment design.** This section shall be divided into the following paragraphs to describe the design of a system or segment.
 - 3.1 **System/segment behavioral design.** This paragraph shall be divided into subparagraphs as needed to describe design decisions that transcend the internal structure of the system. It shall include, as applicable:
 - a. Decisions on allowed/expected inputs/operations, including source, media, volume, frequency, sequence, priority, timing, format, units of measure, range of values, abbreviations, and codes
 - b. Decisions on planned behavior in response to each

input/operation, including sequence, timing, selected equations/algorithms/rules, and handling of illegal inputs

- c. Decisions on system displays, reports, etc., including formats, media, volume, frequency, priority, timing, recipients, forms, security, privacy, units of measure, range of values, abbreviations, and codes
- d. Decisions on system databases/data banks, including contents, size, records/entries, media, security, privacy, retention schedule, and names, synonyms, definitions, units, formats, range, abbreviations, and codes for the data elements
- e. Decisions on the type of flexibility and levels of availability, integrity, and confidentiality to be offered by the system

3.2 System architecture. This paragraph shall be divided into subparagraphs as needed to provide the following information, using diagrams as needed:

- a. Describe the internal structure of the system, identifying the segments, Hardware Configuration Items (HWCI), and CSCIs and summarizing the purpose of each
- b. Describe the relationships among the segments, HWCI, and CSCIs, including sequence of execution, as applicable
- c. Identify, state the purpose, and provide a high-level description of each external interface of the system

3.3 Allocation of requirements. This paragraph shall be divided into subparagraphs as needed to:

- a. Identify the system requirements allocated to each HWCI
- b. Identify the system requirements allocated to each CSCI
- c. Identify the system requirements allocated to each manual operation
- d. Identify the system requirements affecting each internal HWCI-to-HWCI interface
- e. Identify the system requirements affecting each internal HWCI-to-CSCI interface

- f. Identify the system requirements affecting each internal CSCI-to-CSCI interface

3.4 Processing resources. This paragraph be divided into subparagraphs as needed to describe the processing resources to be used for the system. For each processing resource, this paragraph shall specify its hardware, programming, design, coding, and utilization characteristics, including for computer hardware:

- a. Amount of internal memory
- b. Processing speed
- c. Auxiliary storage capacities
- d. Growth capabilities
- e. Diagnostic capabilities
- f. Additional computer hardware capabilities needed to support the software being developed
- g. The allocation of pertinent processing resources to each CSCI

3.5 Requirements traceability. This paragraph shall provide traceability from the system requirements to the HWCIs, CSCIs, and manual operations of the system, to demonstrate that each system requirement has been allocated. The traceability may be shown in a table.

4. Software (CSCI) design. This section shall be divided into the following paragraphs to describe the design of one or more CSCIs.

4.1 (CSCI name and project-unique identifier).

4.1.1 CSCI-wide behavioral design. This paragraph shall be divided into subparagraphs as needed to describe CSCI design decisions that transcend the internal structure of the CSCI. It shall include, as applicable:

- a. Decisions on allowed/expected inputs from users and other sources, including name(s), source, media, volume, frequency, sequence, priority, timing, security, privacy, format, units of measure, range of values, abbreviations, codes, examples
- b. Decisions on planned behavior in response to each input, including actions, sequence, conditional behavior,

timing, selected equations/algorithms/rules, disposition, and handling of illegal or "out-of-bounds" inputs

- c. Decisions on CSCI-produced outputs (displays, reports, etc.), including name(s), format/layout, media, volume, frequency, priority, timing, recipients, forms, security, privacy, units of measure, range of values, abbreviations, codes, disposition, examples
- d. Decisions on CSCI-contained databases/data banks, including (1) contents: size, records/entries, media, security, privacy, retention; (2) data elements: name(s), definitions, units, formats, range, abbreviations, security, privacy, codes for the data elements; (3) relationship to CSCI capabilities/functions; (4) data retention/reporting. Reference may be made to Database Design Documents.
- e. Decisions on the type of flexibility and the levels and types of availability, security, privacy, and continuity of operations to be offered by the CSCI

4.1.2 CSCI design conventions. This paragraph shall explain any rules, schemes, and conventions used in designing the CSCI and expressing that design. Design standards may be referenced. This information shall include, as applicable:

- a. Mnemonic identifiers and their use
- b. Conventions for design diagrams, listings, abbreviations, comments, and symbols appearing in diagrams and listings
- c. Standard data elements and related features

4.1.3 CSCI architectural design. This paragraph shall be divided into subparagraphs as needed to present the following information. The information may be presented graphically, for example in diagrams, tables, or flow charts.

- a. The role of the CSCI within the system architecture, including the purpose of each CSCI-external interface
- b. The internal architecture of the CSCI, including the following, annotated to identify differences for each state and mode in which the CSCI operates:

- 1) The name/ID and purpose of each software unit comprising the architecture
- 2) The structure of the CSCI, including, as applicable:
 - a) The flow of execution control among software units, including any dynamically controlled sequencing during the CSCI's operation
 - b) The flow of data among software units
- 3) Any non-developmental software to be incorporated into the CSCI
- 4) The allocation of memory and processing time to the software units

4.1.4 CSCI detailed design. This paragraph shall be divided into the following subparagraphs to provide a design description of each software unit of the CSCI. The information may be presented graphically, for example in diagrams, tables, or flow charts; in a program design language; or by reference to other design representations including headers of the code.

4.1.4.1 (Software unit name and project-unique identifier). This paragraph shall identify a software unit by name and project-unique identifier. Alternatively, it may identify a group of software units, with each software unit identified in a subparagraph. The design description of each software unit shall include the following, as applicable:

- a. The purpose of the software unit
- b. The requirements allocated to the software unit
- c. The derived design requirements for the software unit (e.g., the algorithms to be incorporated into the software unit)
- d. Any constraints, limitations, or unusual features in the design of the software unit
- e. The programming language and rationale if different from that for the CSCI
- f. Reusable software to be used, where it resides, where its design is documented
- g. The program library in which the software unit is to be placed

- h. The inputs, outputs, and other data elements of the software unit, including, as applicable for each:
 - 1) Project-unique identifier
 - 2) Brief description including purpose
 - 3) Units of measure, such as knots, seconds, meters, dollars
 - 4) Limit/range of values (for constants, the actual value)
 - 5) Accuracy
 - 6) Precision/resolution in terms of significant digits
 - 7) Priority
 - 8) Frequency at which the data element is input, calculated, refreshed, or output
 - 9) Legality checks to be performed on the data element
 - 10) Data type, such as integer, ASCII, fixed, real, enumeration
 - 11) Data representation/format/structure
 - 12) Sources, including other software units where the data element is set
 - 13) Destinations, including other software units where the data element is used
 - 14) Maximum size and storage needs
 - 15) Access method, such as random or sequential
- i. The logic flow to be used by the software unit, including:
 - 1) Conditions under which software unit execution is initiated
 - 2) Conditions under which control is passed to other software units
 - 3) Response and response time to each input
 - 4) Dynamically controlled sequencing during the software unit's operations

4.1.5 Requirements traceability. This paragraph shall provide traceability from the CSCI requirements to the software units. The traceability may be shown in a table.

5. Interface design. This section shall be divided into the following paragraphs to describe the design of one or more CSCI-external interfaces, that is, the specific interface characteristics selected to respond to the CSCI-external interface requirements. If part of all of this information

is documented elsewhere, it may be referenced.

5.1 Interface identification and diagrams. This paragraph shall identify and provide an overview of each interface between the CSCIs to which this document applies and other HWCIs, CSCIs, and critical items. This description may be provided by one or more interface diagrams, as appropriate, and shall characterize each interface (sequential or concurrent operation, real-time data transfer, store-and-retrieve data transfer, operator controlled, etc.)

5.1 (Interface name and project-unique identifier). This paragraph (beginning with 5.2) shall identify an interface by name and project-unique identifier and shall state its purpose. This paragraph shall be divided into the following subparagraphs.

5.1.1 Data elements. This paragraph shall provide, possibly in a data element definition table, the following information, as applicable, for each data element transmitted between the interfacing items:

- a. A project-unique identifier for the data element
- b. A brief description of the data element
- c. The CSCI, HWCI, or other item that is the source of the data element
- d. The CSCI(s), HWCI(s), or other item(s) that are the recipients of the data element
- e. The units of measure of the data element, such as seconds, meters, dollars
- f. The limit/range of values of the data element (for constants, provide the actual value)
- g. The accuracy of the data element
- h. The precision or resolution of the data element in terms of significant digits
- i. The timing characteristics of the data element, for example, how often sent or received, how long transmitted
- j. Legality checks the data element must be able to pass
- k. The data type, such as integer, ASCII, fixed, real, enumerated
- l. The data representation/format
- m. The sequence and other dependencies of the data element

5.1.2 Messages or other data assemblies. This paragraph shall

identify by name and project-unique identifier each message or other assembly of data elements transmitted between the interfacing items, and shall describe the assignment of data elements to each message or assembly. A cross-reference of each message or assembly to the data elements it contains, and vice versa, shall be provided.

5.1.3 Interface priorities. This paragraph shall specify the relative priority of the interface and of each data element, message, or assembly transmitted between the interfacing items.

5.1.4 Interface/communication protocols. This paragraph shall be divided into subparagraphs as needed to describe the commercial, military, or proprietary communications protocols associated with the interface. Each protocol shall be identified and described, including the following, as applicable:

- a. Fragmentation and re-assembly of messages
- b. Message formatting
- c. Legality checks, error control and recovery procedures, including fault tolerance, handling of "out-of-bounds" conditions, and features to ensure continuity of operations in the event of emergencies
- d. Synchronization, including connection establishment, maintenance, termination, and timing
- e. Flow control, including sequence numbering, window size, and buffer allocation
- f. Data transfer rate, whether periodic/aperiodic, minimum interval between transfers
- g. Routing, addressing, and naming conventions
- h. Transmission services, including priority and grade
- i. Status, identification, notification, and any other reporting features
- j. Security and privacy, including encryption, user authentication, compartmentalization, and auditing

6. Database design. This section shall be divided into the following paragraphs to describe the logical and physical design of one or more databases.

6.1 (Database name and project-unique identifier).

6.1.1 Database overview. This paragraph shall be divided into

subparagraphs as needed to provide the following overview information about the database:

- a. Database environment:
 - 1) Identification of systems that will use the database
 - 2) Relationship to other databases, including supersession
 - 3) Estimated internal and peripheral storage requirements, including variations for different modes of operation
 - 4) Mapping of database components to one another and to the files on which they will be recorded
 - 5) Data communications environment in which the database will be installed
- b. Internal labeling conventions used in this database design
- c. Organization of the database and major design considerations:
 - 1) The conceptual model used for the database
 - 2) Physical allocation, including for each area/file:
 - a) General area/file design and format
 - b) Rationale for the design
 - c) Inter-area/file dependencies of the area/file
- d. Instructions to be followed by personnel who will contribute to the generation of the database and who will use it for testing and operational purposes, including, as applicable:
 - 1) Any special criteria for entering data into the database
 - 2) Source documents for the rules/procedures for submitting data for entry
 - 3) Source documents for the machine run instructions for generating, modifying, updating, or otherwise using the database
- e. Support software available for handling the database (such as database analysis/ sizing/loading/repairing software), including name, functions, major operating

considerations, and documentation references

- f. Overview of security, privacy and criticality considerations for the database

6.1.2 Database administrative information. This paragraph shall be divided into subparagraphs as needed to provide the information necessary to establish and administer the database in the described environment. It shall include, as applicable:

- a. Identification of the offices responsible for database administrative functions
- b. System information, including:
 - 1) The identification, vendor, version or release date, and targeted hardware of the Database Management System (DBMS) to be used
 - 2) Hardware configurations on which the database can reside
 - 3) Identification and document references for any DBMS utility software
 - 4) Use and management of integrity and access controls
- c. Schema information, including:
 - 1) Rationale for the chosen database structure
 - 2) Content of the database, listing its data elements and indicating within which subschemas of the database they are visible
 - 3) Description of schema and each subschema of the database, including name, file type and name, Data Description Language, access control keys, concurrence locking, data name mapping, overall area/file limitations and controls, restrictions due to redefinitions and access paths, and any other limitations or restrictions
 - 4) Logical organization of the data into records/tuples/sets/predefined relationships
 - 5) Physical structure (e.g., areas/files, indexes, pointers) and methods for achieving operating efficiency
 - 6) Sizing formulas for determining required storage
 - 7) Recovery methods for reestablishing schema and

support files

8) Cross-reference to applicable requirements

d. Area/file information, including:

1) Rationale for the chosen physical structure

2) Content of each area/file, listing the records it contains and their purposes

3) Description of each area/file, including name, type, code, mapping, limitations and controls, access procedures/mechanisms, and training/testing capabilities

4) Information about data storage and any limiting factors regarding allocation, expansion, paging, loading, size

5) Recovery methods for reconstructing the necessary data and structure

6.1.3 Use of the database by (CSCI or system identification).

This paragraph shall be divided into subparagraphs as needed to provide technical information needed by applications developers to use the database and its utilities in the development, maintenance, or enhancement of software. When more than one CSCI or system will use the database, additional paragraphs similar to this one shall be added. Each shall include, as applicable:

a. Detailed technical descriptions of the database, including the name and purpose of each subschema or local view, and for each:

1) A list and description of each record in the subschema, including name, code, size, other needed attributes, and description of data elements in the record

2) A list and description of each set or predefined relation in the subschema

3) A list and description of any areas/files of the database, including name, code, type, allocation, expansion, load factor, database keys, and page size

4) A list and description of each access routine and query path structure developed for this subschema or local view

- 5) A description of security and privacy controls for the subschema
 - b. Information about utility software to aid in database use for this CSCI or system
 - c. A description of any error handling routines and procedures available
 - d. A list of all messages output to the CSCI during execution of database software
7. **Notes**. This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, rationale). This section shall contain an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document, and a list of any terms and definitions needed to understand this document.
- A. **Appendixes**. Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, etc.).

| | |
|---|--|
| COAST GUARD DATA ITEM DESCRIPTION | APPROVAL DATE: |
| Public reporting burden for collection of this information is estimated to average 110 hours per response, including the time for reviewing instructions, searching for existing data sources, gathering and maintaining the data needed and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to COMMANDANT (G-SCC-2), 2100 Second Street S.W., Room 6410, Washington, DC 20593-0001. | |
| 1. TITLE: Coast Guard Software Test Document (CG-STD) | 2. IDENTIFICATION NUMBER: DI-CGSDDS-004 |
| 3. DESCRIPTION/PURPOSE: 3.1 The Software Test Document (CG-STD) is a single document encompassing the test planning, test cases and procedures, and test results for Computer Software Configuration Item (CSCI), CSCI integration, and software system testing, as applicable. | |
| 4. CLIN NUMBER(s): | 5. OFFICE OF PRIMARY RESPONSIBILITY: USCG Software Policy & Standards Division (G-SCC-2) |
| 6. APPLICATION/INTERRELATIONSHIP: 6.1 This Data Item Description (DID) contains the format and contents preparation instructions for data generated under the work task described in paragraphs 5.1.2, 5.1.3, 5.10.1, 5.10.2, 5.10.6, 5.11.1, 5.11.2, 5.11.6, 5.12.1, 5.12.2, 5.12.6, 5.13.1, 5.13.2 and 5.13.6 of CG-SDDS (COMDTINST 5234.4). 6.2 Commercial, or other existing documents may be substituted for all or part of this document if they contain the required data. | |
| 7. PREPARATION INSTRUCTIONS: 7.1 <u>General Instructions.</u> The document shall be developed according to the following format specifications: a. <u>Automated Techniques.</u> Use of automated techniques is encouraged. The Contract Data Requirements List (CDRL) should specify whether the data are to be delivered on paper or electronic media; any requirements on the electronic representation (such as ASCII, CALS, or compatibility with a specific word processor or other software); whether the data may be delivered in contractor format rather than in the format specified herein; and whether the data may reside in a computer-aided software engineering (CASE) or other automated tool rather than in the form of a traditional document. b. <u>Alternate presentation styles.</u> Charts, tables, matrices, and other presentation styles are acceptable substitutes for text when data required by this DID can be made more readable using these styles. (Continued on Page 2) | |
| 8. DISTRIBUTION STATEMENT: <u>DISTRIBUTION STATEMENT A.</u> This DID will be incorporated into a COMDTINST along with the approved CG Software Development & Documentation Standards (CG-SDDS). | |

7. PREPARATION INSTRUCTIONS (continued)

- c. **Title page or identifier.** When data are delivered in the form of a traditional document, the document shall include a title page containing, as applicable: document number; volume number; version/revision indicator; security markings or other restrictions on the handling of the document; date; document title; name, abbreviation, and any other identifier for the system, subsystem or item to which the document applies; contract number; CDRL item number; organization for which the document has been prepared; name and address of preparing organization. For data delivered in an alternative form, this information shall be included on external and internal labels or by equivalent identification methods.
- d. **Table of contents.** When data are delivered in the form of a traditional document, the document shall contain a table of contents providing the number, title, and page number of each titled paragraph, figure, table, and appendix. For data delivered in an alternative form, this information shall consist of an internal or external table of contents containing pointers to, or instructions for accessing, each paragraph, figure, table, and appendix or their equivalents.
- e. **Page numbering/labeling.** When data are delivered in the form of a traditional document, each page shall contain a unique page number and display the document number, including version, volume, and date, as applicable. For data delivered in an alternative form, files, screens, or other entities shall be assigned names or numbers in such a way that desired data can be indexed and accessed.
- f. **Response to tailoring instructions.** When data are delivered in the form of a traditional document, paragraphs that have been tailored out of the DID shall result in the corresponding paragraph number and title in the document, followed by "This paragraph has been tailored out." For data delivered in an alternative form, this representation need occur only in the table of contents or equivalent.
- g. **Multiple paragraphs and subparagraphs.** Any section, paragraph, or subparagraph in this DID may be written as multiple paragraphs or subparagraphs to enhance readability

7.2 **Content.** Content requirements begin on the following page.

The numbers shown designate the paragraph numbers to be used in the document. Each such number is understood to have the prefix "7.2" within this DID. For example, the paragraph numbered 1.1 is understood to be paragraph 7.2.1.1 within this DID.

1. **Scope.** This section shall be divided into the following paragraphs.
 - 1.1 **Identification.** This paragraph shall contain a full identification of the system and the CSCIs to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s).
 - 1.2 **System overview.** This paragraph shall briefly state the purpose of the system and the software to which this document applies. It shall describe the general nature of the system and software; summarize the history of system development, operation, and maintenance; identify the project sponsor, user, developer, and support agencies; identify current and planned operating sites; and list other relevant documents.
 - 1.3 **Document overview.** This paragraph shall summarize the purpose and contents of this document.
2. **Referenced documents.** This section shall list by document number and title all documents referenced in this document. This section shall also identify the source for all documents not available through normal Government stocking activities.
3. **Software test planning.** This section shall be divided into the following paragraphs to describe the contractor's plans for conducting CSCI, CSCI integration, and software system testing. (Note: When the software development is performed in-house, the term "contractor" will be understood to mean "in-house developer")
 - 3.1 **Software test environment.** This paragraph shall be divided into subparagraphs as needed to describe the software test environment at each intended test site. Reference may be made to the software engineering environment described in the software development plan for those resources that are used in both environments.
 - 3.1.1 (Name of test site(s)). This paragraph shall identify one or more test sites to be used for the testing, and shall

describe the software test environment at the site(s). The description of each test environment shall include the following:

- a. Name/identifier and version of software items (e.g., operating systems, compilers, communications software, databases, test drivers, preprocessors, other special test software), including the purpose of each, any security and privacy considerations, and which are to be provided by the site
- b. Name/model/serial number and version of hardware and firmware items (e.g., computer hardware, interfacing equipment, communications equipment, test message generators, test timing devices, etc.), including the purpose of each, any security and privacy considerations, and which are to be provided by the site
- c. Other materials needed for the testing (e.g., manuals, listings, the software to be tested, data to be used in the tests, sample outputs, test worksheets, etc.), including the purpose of each, any security and privacy considerations, and which are to be provided by the site
- d. The proprietary nature and Government rights associated with each item of the software test environment
- e. Plans for installation, testing, and control of the software test environment
- f. The organizations that will participate in the testing
- g. The number and skill types of personnel required during the test period, and the dates and time they will be needed
- h. Plans for any orientation and training to be given before and during the testing
- i. The tests to be performed at the site(s), indicated by references to paragraph 3.2

3.2 Test identification.

3.2.1 General information. This paragraph shall provide the

following general information about the testing to be performed:

- a. Levels of tests to be performed (e.g. CSCI, CSCI integration, system)
- b. Types or classes of tests to be performed (e.g., stress tests, timing tests, erroneous input tests, maximum capacity tests)
- c. Any test requirements that apply to all, or a group of, tests (e.g., "Tests of type x shall use live data")
- d. Extent of testing to be performed (e.g., specified sampling of possible inputs) and rationale for the extent selected
- e. Planned sequence or progression of tests
- f. The data recording, reduction, and analysis procedures to be used during and after the tests

3.2.2 Planned tests.

3.2.2.1 (Item(s) to be tested). This paragraph shall identify a CSCI, group of CSCIs, subsystem, system, or other entity by name and project-unique identifier, and shall be divided into the following subparagraphs to provide an overview of the tests planned for that item. (Note: "tests" are collections of test cases. There is no intent to describe each test case in this paragraph.)

3.2.2.1.1 (Test name and identifier). This paragraph shall identify a test by name and project-unique identifier, and shall be divided into subparagraphs as needed to specify:

- a. Test objective
- b. Test level (using the test levels defined above)
- c. Test class (using the test classes defined above)
- d. Qualification method(s) specified in the associated requirements specification(s)
- e. Cross reference to the software requirements addressed by this test
- f. Any special test requirements
- g. Type of data to be recorded

- h. Type of data recording/reduction/analysis to be employed
- i. Assumptions and constraints
- j. Any security and privacy considerations

3.3 Test schedules. This paragraph shall contain or reference the schedules for conducting the tests identified in this plan. The schedules shall include:

- a. Time frame for testing at each test site
- b. Test schedule for each test site, including pretest activities, preparation of test inputs and databases, any orientation or training associated with the testing, the testing itself, and any period for review/approval of test results

4. Software test descriptions. This section shall be divided into the following paragraphs to describe the test cases and test procedures to be used for CSCI, CSCI integration, and software system testing.

4.1 (Item(s) to be tested). This paragraph shall identify an item to be tested and shall be divided into the following subparagraphs to describe the tests for that item.

4.1.1 Test preparations. This paragraph shall be divided into the following subparagraphs to describe test preparations. Security and privacy considerations shall be included as applicable.

4.1.1.1 (Test name and identifier). This paragraph shall identify a test by name and identifier, and shall be divided into subparagraphs as needed to provide the following information:

- a. Test schedule, if updated or more detailed than that given in paragraph 3.3
- b. Pre-test procedures for hardware preparations
- c. Pre-test procedures for software and data preparations
- d. Other pre-test preparations

4.1.2 Test descriptions. This paragraph shall be divided into the following subparagraphs to provide test descriptions for the item(s) under test. Security and privacy considerations shall be included as applicable.

4.1.2.1 (Test name and identifier). This paragraph shall identify a test by name and project-unique identifier and shall be divided into the following subparagraphs.

4.1.2.1.1 (Test case name and project-unique identifier). This subparagraph shall identify a test case by name and project-unique identifier and shall provide the following information for the test case:

- a. Purpose/traceability: cross-reference to the software requirements addressed by the test case
- b. Means of control to be used: manual, semi-automatic, or automatic insertion of inputs and control of test sequence
- c. Prerequisite conditions that must be established: hardware/software initialization or other initialization
- d. Test inputs: description, source, real vs simulated, timing/sequence, means of control
- e. Expected results, both intermediate and final, as applicable
- f. Criteria for evaluating the results: acceptable limits for tolerances, sample sizes, durations, number of breaks, severity of problems, status indicators
- g. Test procedure: numbered steps to be followed in initiating, carrying out, and analyzing the results of the test case, including, as applicable, alternative actions, expected results, and evaluation criteria for each step
- h. Any assumptions or constraints applicable to the test case

5. Software test report. This section shall be divided into the following paragraphs to describe the results of CSCI, CSCI integration, and software system testing.

5.1 (Item(s) tested). This paragraph shall identify an item under test and shall be divided into the following subparagraphs to describe the results of the tests for that

item.

5.1.1 Test overview.

5.1.1.1 (Test name and identifier). This paragraph shall identify a test by name and project-unique identifier and shall be divided into subparagraphs as needed to provide the following information:

- a. A summary of test results, including the completion status of the test (for example, "all results as expected," "problems encountered," "deviations required") and, when the completion status is not "as expected," a summary of the problems or deviations that occurred and reference Section to 5.x.2 for details
- b. Reference to a figure, appendix, or other presentation of a chronological record of test events, including dates, times, locations, hardware and software configurations used, activities, and test performers and witnesses, as applicable

5.1.2 Detailed test results. This paragraph shall be divided into the following subparagraphs to describe the detailed results of each test.

5.1.2.1 (Test name and identifier). This paragraph shall identify a test by name and project-unique identifier and shall be divided into subparagraphs as needed to provide the following information:

- a. The completion status of each test case associated with this test (e.g., "all results as expected," "problems encountered," "deviations required")
- b. A description of each problem encountered, including the test case and test procedure step where it was encountered and reference to the associated problem report(s)
- c. A description of each deviation from the documented test cases/test procedures, including test case and test procedure step where the deviation occurred, nature of the deviation, rationale, and an assessment of its impact on the validity of the test case

- 5.1.3 Evaluation and recommendations.** This paragraph shall be divided into the subparagraphs as needed to provide:
- a. An overall assessment of each item under test, summary of remaining deficiencies or limitations, impact if the deficiencies/limitations are not corrected, and impact to correct them
 - b. An assessment of the manner in which the test environment may be different from the operational environment and the effect of this difference on the capabilities tested
 - c. Any recommended improvements in the design, operation, or testing of the item(s) under test (optional)
- 6. Notes.** This section shall contain any general information that aids in understanding this document (e.g., background information, glossary). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.
- A. Appendixes.** Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, etc.).

| | |
|---|--|
| COAST GUARD DATA ITEM DESCRIPTION | APPROVAL DATE: |
| Public reporting burden for collection of this information is estimated to average 110 hours per response, including the time for reviewing instructions, searching for existing data sources, gathering and maintaining the data needed and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to COMMANDANT (G-SCC-2), 2100 Second Street S.W., Room 6410, Washington, DC 20593-0001. | |
| 1. TITLE: Coast Guard Software User/Operator Manual (CG-SUOM) | 2. IDENTIFICATION NUMBER: DI-CGSDDS-005 |
| 3. DESCRIPTION/PURPOSE: 3.1 The Software User/Operator Manual (CG-SUOM) is a single document encompassing all instructions to hands-on users, computer center software operators, preparers and users of software input/output, and operators of computer hardware. | |
| CLIN NUMBER(s): | 5. OFFICE OF PRIMARY RESPONSIBILITY: USCG Software Policy & Standards Division (G-SCC-2) |
| 6. APPLICATION/INTERRELATIONSHIP: 6.1 This Data Item Description (DID) contains the format and contents preparation instructions for data generated under the work task described in paragraphs 5.14.1, 5.14.2, 5.14.3, and 5.14.4 of CG-SDDS (COMDTINST 5234.4). 6.2 Commercial, or other existing documents may be substituted for all or part of this document if they contain the required data. | |
| 7. PREPARATION INSTRUCTIONS: 7.1 <u>General Instructions.</u> The document shall be developed according to the following format specifications: a. <u>Automated Techniques.</u> Use of automated techniques is encouraged. The Contract Data Requirements List (CDRL) should specify whether the data are to be delivered on paper or electronic media; any requirements on the electronic representation (such as ASCII, CALS, or compatibility with a specific word processor or other software); whether the data may be delivered in contractor format rather than in the format specified herein; and whether the data may reside in a computer-aided software engineering (CASE) or other automated tool rather than in the form of a traditional document. b. <u>Alternate presentation styles.</u> Charts, tables, matrices, and other presentation styles are acceptable substitutes for text when data required by this DID can be made more readable using these styles. <p style="text-align: right;">(Continued on Page 2)</p> | |
| 8. DISTRIBUTION STATEMENT: DISTRIBUTION STATEMENT A. This DID will be incorporated into a COMDTINST along with the approved CG Software Development & Documentation Standards (CG-SDDS). | |

7. PREPARATION INSTRUCTIONS (continued)

- c. **Title page or identifier.** When data are delivered in the form of a traditional document, the document shall include a title page containing, as applicable: document number; volume number; version/revision indicator; security markings or other restrictions on the handling of the document; date; document title; name, abbreviation, and any other identifier for the system, subsystem or item to which the document applies; contract number; CDRL item number; organization for which the document has been prepared; name and address of preparing organization. For data delivered in an alternative form, this information shall be included on external and internal labels or by equivalent identification methods.
- d. **Table of contents.** When data are delivered in the form of a traditional document, the document shall contain a table of contents providing the number, title, and page number of each titled paragraph, figure, table, and appendix. For data delivered in an alternative form, this information shall consist of an internal or external table of contents containing pointers to, or instructions for accessing, each paragraph, figure, table, and appendix or their equivalents.
- e. **Page numbering/labeling.** When data are delivered in the form of a traditional document, each page shall contain a unique page number and display the document number, including version, volume, and date, as applicable. For data delivered in an alternative form, files, screens, or other entities shall be assigned names or numbers in such a way that desired data can be indexed and accessed.
- f. **Response to tailoring instructions.** When data are delivered in the form of a traditional document, paragraphs that have been tailored out of the DID shall result in the corresponding paragraph number and title in the document, followed by "This paragraph has been tailored out." For data delivered in an alternative form, this representation need occur only in the table of contents or equivalent.
- g. **Multiple paragraphs and subparagraphs.** Any section, paragraph, or subparagraph in this DID may be written as multiple paragraphs or subparagraphs to enhance readability

7.2 **Content.** Content requirements begin on the following page.

The numbers shown designate the paragraph numbers to be used in the document. Each such number is understood to have the prefix "7.2" within this DID. For example, the paragraph numbered 1.1 is understood to be paragraph 7.2.1.1 within this DID.

1. **Scope.** This section shall be divided into the following paragraphs.
 - 1.1 **Identification.** This paragraph shall contain a full identification of the system, CSCIs, and computer system(s) to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s).
 - 1.2 **System overview.** This paragraph shall briefly state the purpose of the system, CSCIs, and computer system(s) to which this document applies. It shall describe the general nature of the system and software; summarize the history of system development, operation, and maintenance; identify the project sponsor, user, developer, and support agencies; identify current and planned operating sites; and list other relevant documents.
 - 1.3 **Document overview.** This paragraph shall summarize the purpose and contents of this document.
 - 1.4 **Security and privacy.** This paragraph shall contain an overview of the security and privacy considerations associated with the system. A warning shall be included regarding making unauthorized copies of data, software, or documents if applicable.
2. **Referenced documents.** This section shall list by document number and title all documents referenced in this document. This section shall also identify the source for all documents not available through normal Government stocking activities.
3. **Software user manual.** This section shall be divided into the following paragraphs to tell a hands-on software user how to install, initiate, use, and terminate the software in a single CSCI, a group of related CSCIs, or an overall software system. If the software has no hands-on users, this section shall state "Not applicable."
 - 3.1 **Software summary.** This paragraph shall be divided into subparagraphs as needed to provide a nontechnical description of the software from the point of view of the hands-on user. The description shall include:

- a. An overview describing the use of the software in supporting the activities of the user, the functions performed, logical parts and communication paths from a user view, interfaces with other systems and with user organizations, performance capabilities that can be expected, and supervisory controls that can be implemented to manage the software
- b. The software environment, including hardware and software that must be present for the software described in this manual to run
- c. Contingencies and alternate modes of operation, including differences in what the user will be able to do with the software at times of emergency and in peacetime, war, and conditions of alert
- d. Points of contact and procedures to be followed to obtain assistance and report problems encountered in using the software

3.2 Access to the software. This paragraph shall be divided into subparagraphs as needed to provide information oriented to the first time/occasional user. The information shall include:

- a. Information for a first-time user, including:
 - 1) Familiarization with the equipment (such as power on, power off, and characteristics of the visual display screen, cursor, and keyboard)
 - 2) An overview of access, security, and privacy features of the software, such as obtaining a password, adding/changing/deleting passwords, and security and privacy considerations
 - 3) Any special procedures that the user must perform to be identified or authorized to access or install software on the equipment, or to enter parameters for software operation
- b. Step-by-step procedures for beginning work, including any options available and a checklist for problem determination

- c. Description of how to cease or interrupt use of the software and how to determine whether normal termination or cessation has occurred

3.3 Processing reference guide. This paragraph shall be divided into subparagraphs to provide detailed processing procedures. If procedures are complicated or extensive, additional paragraphs 3.4 - 3.n may be added using the same paragraph structure. The procedures may be stated in terms of functions, menus, screens, transactions, or other processes that best describe the software to the user.

3.3.1 Capabilities. This paragraph shall briefly describe the interrelationships of the transactions, menus, functions, or other processes in order to provide an overview of the use of the software.

3.3.2 Conventions. This paragraph shall describe any conventions such as the use of colors in displays, the use of audible alarms, the use of abbreviated vocabulary, and the use of rules for assigning names or codes.

3.3.3 Processing procedures. This paragraph shall be divided into subparagraphs as needed to present the processing procedures. The subparagraphs may be organized by function, menu, transaction, or other method, as appropriate for the software. The subparagraphs shall describe and give options and examples, as applicable, of menus, data entry forms, user inputs, inputs from other software or hardware that may affect the software's interface with the user, outputs, diagnostic or error messages or alarms, and help facilities that can provide on-line descriptive or tutorial information.

3.3.4 Related processing. This paragraph shall identify and describe any related batch, offline, or background processing performed by the software that is not invoked directly by the user and is not described in paragraph 3.3.3. Any user responsibilities to support this processing shall be specified.

3.3.5 Data backup. This paragraph shall describe procedures for creating and retaining backup data that can be used to replace primary copies of data in event of errors, defects, malfunctions, or accidents.

- 3.3.6 Recovery from errors, malfunctions, and emergencies.** This paragraph shall present detailed procedures for restart or recovery from errors or malfunctions occurring during processing and for ensuring continuity of operations in the event of emergencies.
- 3.3.7 Messages.** This paragraph shall list, or refer to an appendix that lists, all error messages, diagnostic messages, and information messages that can occur while accomplishing any of the user's functions. The meaning of each message and the action that should be taken after each such message shall be identified and described.
- 4. Computer center software operator manual.** This section shall be divided into the following paragraphs to provide personnel in a computer center a detailed operational description of a software system, its operating environment, and procedures for performing computer runs of the system. If the software will not operate in a computer center, this section shall state "Not applicable."
- 4.1 System description.** This paragraph shall be divided into subparagraphs as needed to provide a nontechnical description of the software from the point of view of the computer center software operator. The description shall include:
- a. System application: the intended uses of the system
 - b. System organization: a chart showing the data processing operations of the system, including how the different operations are interrelated (for example runs grouped by time periods, cycles, or organizational levels)
 - c. Software inventory: an inventory of the software that makes up the system, including the software's full name, identification, security and privacy considerations, and identification of the software necessary to continue or resume operation in case of an emergency
 - d. Resource inventory: a list of permanent files and databases/data banks that are referenced, created, or updated by the system, including names, file

identifications, storage media, required storage (number of tapes or disks), security and privacy considerations, and identification of those files and databases/data banks necessary to continue or resume operation in case of an emergency

- e. Report inventory: a list of all reports produced by the system, including report names and the software that produces each report
- f. Processing overview: an overview of system processing, including system restrictions, waivers of operational standards, information oriented toward specific support areas (e.g., library, small computer and teleprocessing support), or other processing requirements such as interfaces with other systems or other pertinent system-related information
- g. Communications overview: a general description of the communications functions and processes of the system, including a chart of the communications network involved within the system
- h. Security, privacy, and continuity of operations: an overview and discussion of the security and privacy considerations associated with the system, and the actions necessary to continue or resume operation in case of an emergency

4.2 Description of runs. This paragraph shall provide a description of the runs for use in scheduling of operations, assignment of equipment, management of input and output data, and restart/recovery procedures. In online systems some information about system operational control may be related to the capabilities of the operating system. Much of the information may be included in graphic representations.

4.2.1 Run inventory. This paragraph shall provide a list of the runs showing the software and the jobs that make up each run. It shall include a brief summary of the purpose of the run. This list shall relate to the runs that are included in the remainder of this paragraph.

4.2.2 Phasing. This paragraph shall provide a schedule of

acceptable phasing of the system into a logical series of operations. A system run may be phased to permit manual or semiautomatic checking of intermediate results, to provide the user with intermediate results for other purposes, or to permit a logical break if higher priority jobs are submitted. An example of the minimum division for most systems would be edit, file update, and report preparation.

4.2.3 Diagnostic procedures. This paragraph shall furnish the setup and execution procedures for any software diagnostics. Included shall be procedures for validation and trouble shooting. All parameters (both input and output), codes, and range values for diagnostic software shall be explained.

4.2.4 Error messages. This paragraph shall list all error messages output by the system, along with the corresponding correction procedure for each message.

4.2.5 Run description for (Run name or identification). Paragraphs 4.2.5 through 4.2.n shall provide detailed information needed to execute runs of the system, including:

- a. Control inputs: a listing of the runstream of job control statements needed to initiate the run
- b. Run management information, for example, run identification, peripheral and resource needs, security and privacy, method of initiation (such as: on request, as a result of another run, at a predetermined time, etc.), estimated run time, required turnaround time, messages and responses, procedures for taking check points, waivers from operational standards, and contacts for problems experienced with the run
- c. Input-output files: information about the files and databases/data banks that serve as input to or that are created or updated by the run. Included for each shall be name, security and privacy, recording medium, retention schedule, and disposition
- d. Output reports: information about the reports that are produced during the run, including report identification, security and privacy, media, volume of report, number of copies, and distribution of copies

- e. Reproduced output reports: information about computer-generated reports that are subsequently reproduced by other means, including report identification, security and privacy, reproduction technique, paper size, binding method, number of copies, and distribution of copies
 - f. Procedures to be followed concerning restart/recovery in the event of a system failure and continuity of operations in the event of emergencies
5. **Software input/output manual.** This section shall be divided into the following paragraphs to tell a user how to submit queries and other inputs to a batch or interactive software system that is run by computer center staff, and to describe the outputs that can be expected from the system. If the software will not be used in this manner, this section shall state "Not applicable."
- 5.1 **System summary.** This paragraph shall be divided into subparagraphs as needed to provide a nontechnical overview of the software from the point of view of a user who submits queries and other inputs to a batch or interactive software system run by computer center staff, and receives the resulting outputs. The overview shall include the following:
- a. **System description:** the uses for which the system is intended, capabilities, operating improvements, benefits expected from its use, major functions performed by the system, logical parts of the system from the point of view of the user, communication paths and techniques, interfaces and relationships to other systems, and the organizations that provide input to the system or that receive output from it
 - b. **System operation:** the relationships of the functions performed by the system with the organizations or stations that are sources of input to the system and those that are recipients of outputs from it
 - c. **System configuration:** overview of the equipment, communications, and networks used by the system
 - d. **System organization:** relationships among logical parts of the system and a brief description of the role of each part

in the operation of the system

- e. **System performance:** overall performance capabilities that can be expected by the user and constraints, such as capacity limitations or time needed to accomplish major functions. Performance measures may be represented in terms of types, volumes, rate of inputs accepted; types, volume, accuracy, rate of outputs; response time and factors that affect it; limitations on input size, format, queries, or language; capabilities for detecting and correcting errors; typical processing time; provisions allowing extension of system usage; and provisions for alternate processing.
- f. **Contingencies/alternate modes of operation:** differences in what the user will be able to do with the system at times of emergencies, disasters, accidents, peacetime, war, and conditions of alert
- g. **Database/data bank:** method used to store and maintain the data; information on the particular Database Management System (DBMS) used, including the types and usage of the data; and for systems using a data bank, identification of all files that make up the applications system, including file identification, retention, media, and sensitivity
- h. **General description of inputs, processing, outputs:**
 - 1) **Inputs:** the purpose and content of inputs, associated inputs required by the system, source or preparer of the inputs, database/data bank where the input is recorded, security and privacy, and any additional remarks of general information
 - 2) **Processing:** the relationship of the input to the output with a general description of the flow of data through the processing cycle
 - 3) **Outputs:** a list of outputs and their relationship to the inputs, purpose of each output, content of output, associated outputs, distribution (recipients) of outputs, security and privacy, and any additional items of general information

5.2 **Functions related to technical operations.** This paragraph shall provide the details necessary to prepare inputs to the

system and shall explain in detail the characteristics and meaning of the information the system produces as outputs. For use of terminals, this paragraph shall reference paragraph 5.5. The following paragraphs shall detail the procedures to be followed for batch processing runs.

- 5.2.1 Initiation procedures.** This paragraph shall detail the procedures that must be followed to initiate system operation. Included may be information such as sample job request forms or sample control statements. If these procedures are standard or are detailed in another manual, that manual shall be referenced.
- 5.2.2 Description of inputs.** This paragraph shall be divided into subparagraphs as needed to describe the user-prepared inputs to the software. The description shall include, as applicable:
- a. Input conditions:** conditions to be observed in preparing each type or class of input; for example, reason for input, its frequency, origin of the input, medium of input, associated inputs required at the same time, and any other information, such as other recipients of the inputs, priority, security, and privacy considerations
 - b. Input formats:** the layout formats to be used for system inputs and the information that may be entered in the various paragraphs and lines of each format. The explanation of each entry provision shall be keyed to the sample formats shown.
 - c. Composition rules:** rules and conventions that must be observed to prepare the inputs, including input transaction length, format conventions, use of labels to denote data sets, sequencing of inputs, spacing and use of symbols, and forbidden characters or parameter sets
 - d. Input vocabulary:** legal character combinations or codes that must be used to identify or compose input items. An appendix may be provided containing an ordered listing of item codes (and their meaning) that can be entered into an input to the system.
 - e. Examples** that illustrate and explain each type or class of input acceptable by the system, including headers, text,

trailers, permissible omissions, and repeatable portions, as applicable

5.2.3 Description of outputs. This paragraph shall be divided into subparagraphs as needed to describe the outputs of the software. The description shall include, as applicable:

- a. General description:** For each type or class of output, the purpose, frequency, options available, media (physical form) of the output, where the output will appear, and any additional characteristics, such as priority, security, privacy, and associated outputs
- b. Output formats:** the layout of each type or class of output; how to interpret, as applicable, security/privacy markings, header information, information that may appear in the body of the output, trailer information, and additional characteristics such as the meaning of special symbols
- c. Sample outputs:** illustrations of each type or class of output available from the system; the purpose of each output and a detailed description, including, as applicable, the meaning, use, source, and characteristics of each information variable
- d. Output vocabulary:** any codes or abbreviations that appear in the output in a form different from those used in the input description, and their meanings

5.2.4 Use of system outputs. This paragraph shall explain the use of the output by the operational area or activity that receives it.

5.2.5 Recovery & error correction procedures. This paragraph shall list the error codes generated by the software, give their meanings, and describe the corrective actions to be taken by the user. Also included shall be the procedures to be followed by the user with respect to restart, recovery, and continuity of operations in the event of emergencies.

5.2.6 Communications diagnostics. This paragraph shall describe the diagnostic procedures available to the user for validating communications and for identifying and classifying problems.

- 5.3 File query procedures.** This paragraph shall be prepared for a system with a file query retrieval capability that does not use terminals. The instructions necessary for recognition, preparation, and processing of a query applicable to the database/data bank shall be explained in detail in the subparagraphs that follow.
- 5.3.1 System query capabilities.** This paragraph shall illustrate in tabular form the preprogrammed query capabilities provided by the system with a description of, or a cross-reference to, a query code. An example is shown in Figure 1 at the conclusion of this DID.
- 5.3.2 Database/data bank format.** This paragraph shall illustrate the database/data bank format and content. An example is shown in Figure 2. The following information, as applicable, shall be listed for each data element: data element name, synonymous names, definition, format, range and enumeration of values, unit of measurement, and data item names, abbreviations, and codes. When the information is published in a data element dictionary, reference to an entry in the dictionary shall be made rather than including an extract from that dictionary. Any variations in either the inputs or outputs from the format or data items that are used on the database/data bank shall be specifically identified.
- 5.3.3 Query preparation.** This paragraph shall provide instructions for the preparation of any necessary query parameters. Figure 3 shows an example of this format. The details of query input preparation in the context of each specific database/data bank and system retrieval capability shall be repeated as necessary in the form of positive instructions. In cases when the retrieval capability is part of a support system and query input formats are not needed, the specific query statement required shall be listed. Figure 4 shows a specific query statement. The formats provided shall be usable to transcribe queries into technical phrasing of the retrieval system. Subparagraphs shall be used to describe different types of queries.
- 5.3.4 Control instructions.** This paragraph shall provide instructions for the sequencing of runs and for the software necessary to extract responses to query requests from the database/data bank. These instructions shall include the

requirements for, and the preparation of, control statements that may be required by the computer system or software. If extensive information regarding control statement preparation is contained in support system documentation, this documentation may be referenced.

- 5.4 User terminal processing procedures.** This paragraph shall provide the user with technical information on the use of terminals to accomplish processing. If procedures are complicated or extensive, additional paragraphs 5.5 through 5.n may be added in the same paragraph structure as this paragraph.
- 5.4.1 Available capabilities.** This paragraph shall describe in general terms capabilities for retrieval, display, and update of data through terminal operations. This description shall include estimates of the frequency of these operations and identification of the events that cause them to be initiated.
- 5.4.2 Access procedures.** This paragraph shall present the sequence of steps required to initiate system operations and to access the database/data bank. Included shall be such information as the name of the system or subsystem being called and other control information such as the offices or personnel authorized to retrieve or update, time periods when such access is allowed, and information for ensuring that only authorized access is allowed.
- 5.4.3 Display, updates, and retrieval procedures.** This paragraphs shall be divided into subparagraphs to describe the step-by-step procedures necessary to produce the various displays, updates, and retrievals that are available through the use of a terminal. For each procedure, information such as the name of the operation, input formats, and sample responses may be included.
- 5.4.4 Recovery and error correction procedures.** This paragraph shall provide error codes and messages that may be displayed by the system and indicate their meanings and any corrective actions that should be taken. Also included shall be any procedures to be followed by the user with respect to restart, recovery, and continuity of operations in the event of emergencies.

5.4.5 **Termination procedures.** This paragraph shall present the sequence of steps necessary to terminate the processing.

6. **Computer system operator manual.** This section shall identify a computer system by stating the name, identification number, version, and any other identifying information for the computer(s), peripheral equipment, and other constituent parts, and shall be divided into the following paragraphs to provide information for operating and performing diagnostics on the computer system. If this information is provided in commercial or other manuals, a reference to those manuals shall suffice for this section. If there are multiple computer systems to describe, sections 7 - n may be used for this purpose.

6.1 **Computer system operation.**

6.1.1 **Computer system preparation and shutdown.** This paragraph shall be divided into subparagraphs as needed to describe:

- a. Procedures to power-on and power-off the computer system
- b. Procedures to initiate operation of the computer system, including, as applicable:
 - 1) Equipment setup and procedures for pre-operation
 - 2) Procedures to bootstrap the computer system and load software and data
 - 3) Commands typically used during computer system initiation
 - 4) Procedures to initialize files, variables, or other parameters
- c. Shutdown procedures necessary to save data files and other information and to terminate computer system operation.

6.1.2 **Operating procedures.** This paragraph shall be divided into subparagraphs as needed to describe procedures for operating the computer system. If more than one mode of operation is available, procedures for each mode shall be provided. The description shall include:

- a. **Input and output procedures:** input and output media (e.g., magnetic tape, disk, cartridge, etc.) and procedures

required to read and write on these media

- b. **Monitoring procedures:** procedures to be followed for monitoring the system in operation, including trouble and malfunction indications, evaluation techniques for fault isolation, conditions requiring computer system shutdown, and procedures for on-line intervention, abort, and user communications
- c. **Recovery procedures:** automatic and manual procedures to be followed for each trouble occurrence, including steps to be taken to restart computer system operation after an abort or interruption of operation and procedures for recording information concerning a malfunction
- d. **Off-line procedures:** procedures required to operate all relevant off-line equipment/ routines of the computer system.
- e. **Any additional procedures** (e.g., computer system alarms, security and privacy considerations, switch over to a redundant computer system or other measures to ensure continuity of operations in the event of emergencies).

6.2 **Diagnostic features.** This paragraph shall be divided into subparagraphs as needed to describe the following:

- a. A summary of the error-detection and diagnostic features available in the computer system, including error message syntax and hierarchy for fault isolation
- b. A description of each diagnostic procedure to be followed, including the hardware, software, or firmware necessary for executing the procedure; step-by-step instructions for executing the procedure; and diagnostic messages and the corresponding required actions
- c. A description of available diagnostic tools and their application

7. **Notes.** This section shall contain any general information that aids in understanding this document (e.g., background information, glossary). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any

terms and definitions needed to understand this document.
If section 6 has been expanded into section(s) 7, ...n, this section shall be numbered as the next section following section n.

- A. **Appendixes.** Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, etc.).

| Preprogrammed Query Capabilities | |
|--|------------|
| DESCRIPTION | QUERY CODE |
| Number of employees within an organization | A |
| Number of employees in a specific pay grade | B |
| Total gross pay for employees within an organization | C |
| State tax year-to-date for specific state | D |
| FICA tax year-to-date for a specific employee | E |
| Total deductions for a specific employee | F |
| Net pay for a specific employee | G |

Figure 1. Example of preprogrammed query capability.

| Format of Data Record | | | |
|-----------------------|--------|-----------------|---------------------|
| ITEM NAME | FORMAT | RANGE OF VALUES | UNIT OF MEASUREMENT |
| ORG-NAME | 30 A/N | 1-9, A-Z | |
| ORG-ID | 6 A/N | 1-9, A-Z | |
| SOC-SEC-NO | 9 A/N | 0-9 | |
| NAME | 20 A/N | --- | |
| PAY-GRADE | 4 A/N | --- | |
| GROSS-PAY | 6 SN | 0-9 | Dollars |
| GROSS-PAY-YTD | 8 SN | 0-9 | Dollars |
| FED-TAX | 6 SN | 0-9 | Dollars |
| FED-TAX-YTD | 8 SN | 0-9 | Dollars |
| FICA | 6 SN | 0-9 | Dollars |
| FICA-YTD | 8 SN | 0-9 | Dollars |
| STATE-TAX | 6 SN | 0-9 | Dollars |
| STATE-TAX-YTD | 8 SN | 0-9 | Dollars |
| STATE-TAX-CODE | 2 A/N | B-3-F6 | |
| ALLOTMENTS | 6 SN | 0-9 | Dollars |
| NET-PAY | 6 SN | 0-9 | Dollars |
| A/N = Alpha-Numeric | | | |
| SN = Signed Numeric | | | |

Figure 2. Example of data record format.

Figure 1. Example of preprogrammed query capability
 &
 Figure 2. Example of data record format

| Format of Query A | | | |
|--|------------------------------|----|---|
| NUMBER OF EMPLOYEES WITHIN AN ORGANIZATION | | | |
| QUERY ITEM <u>TITLE</u> | CHARACTER <u>POSITION</u> | | CONTENT/ <u>COMMENT</u> |
| Query Designator | 1 | Q | Constant |
| File Number | 2-3 | 01 | Constant |
| Query Number | 4-5 | | Insert 01-99 |
| Security Classification | 10 | U | Unclassified |
| Query Card Format Code | 12 | A | |
| Organization | 14-19 | | Insert ORG-ID as requested by query. Refer to applicable code. |

Figure 3. Example of query format.

| Query Statement |
|---|
| Request - No. of employees within an organization (Office of Secretary of Defense) |
| Query Statement - IF ORG-ID EQ OSD LIST NO OF EMPLOYEES |

Figure 4. Example of query statement.

Figure 3. Example of query format
&
Figure 4. Example of query statement

| | |
|---|--|
| COAST GUARD DATA ITEM DESCRIPTION | APPROVAL DATE: |
| Public reporting burden for collection of this information is estimated to average 110 hours per response, including the time for reviewing instructions, searching for existing data sources, gathering and maintaining the data needed and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to COMMANDANT (G-SCC-2), 2100 Second Street S.W., Room 6410, Washington, DC 20593-0001. | |
| 1. TITLE: Coast Guard Software Support Document (CG-SSD) | 2. IDENTIFICATION NUMBER: DI-CGSDDS-006 |
| 3. DESCRIPTION/PURPOSE: 3.1 The Software Support Document (CG-SSD) is a single document encompassing the version description, product specification, and support manuals for one or more Computer Software Configuration Items (CSCIs). | |
| CLIN NUMBER(s): | 5. OFFICE OF PRIMARY RESPONSIBILITY: USCG Software Policy & Standards Division (G-SCC-2) |
| 6. APPLICATION/INTERRELATIONSHIP: 6.1 This Data Item Description (DID) contains the format and contents preparation instructions for data generated under the work task described in paragraphs 5.14.5, 5.15.3, and 5.15.4 of CG-SDDS (COMDTINST 5234.4). 6.2 Commercial, or other existing documents may be substituted for all or part of this document if they contain the required data. | |
| 7. PREPARATION INSTRUCTIONS: 7.1 <u>General Instructions.</u> The document shall be developed according to the following format specifications: a. <u>Automated Techniques.</u> Use of automated techniques is encouraged. The Contract Data Requirements List (CDRL) should specify whether the data are to be delivered on paper or electronic media; any requirements on the electronic representation (such as ASCII, CALS, or compatibility with a specific word processor or other software); whether the data may be delivered in contractor format rather than in the format specified herein; and whether the data may reside in a computer-aided software engineering (CASE) or other automated tool rather than in the form of a traditional document. b. <u>Alternate presentation styles.</u> Charts, tables, matrices, and other presentation styles are acceptable substitutes for text when data required by this DID can be made more readable using these styles. (Continued on Page 2) | |
| 8. DISTRIBUTION STATEMENT: <u>DISTRIBUTION STATEMENT A.</u> This DID will be incorporated into a COMDTINST along with the approved CG Software Development & Documentation Standards (CG-SDDS). | |

7. PREPARATION INSTRUCTIONS (continued)

- c. **Title page or identifier.** When data are delivered in the form of a traditional document, the document shall include a title page containing, as applicable: document number; volume number; version/revision indicator; security markings or other restrictions on the handling of the document; date; document title; name, abbreviation, and any other identifier for the system, subsystem or item to which the document applies; contract number; CDRL item number; organization for which the document has been prepared; name and address of preparing organization. For data delivered in an alternative form, this information shall be included on external and internal labels or by equivalent identification methods.
- d. **Table of contents.** When data are delivered in the form of a traditional document, the document shall contain a table of contents providing the number, title, and page number of each titled paragraph, figure, table, and appendix. For data delivered in an alternative form, this information shall consist of an internal or external table of contents containing pointers to, or instructions for accessing, each paragraph, figure, table, and appendix or their equivalents.
- e. **Page numbering/labeling.** When data are delivered in the form of a traditional document, each page shall contain a unique page number and display the document number, including version, volume, and date, as applicable. For data delivered in an alternative form, files, screens, or other entities shall be assigned names or numbers in such a way that desired data can be indexed and accessed.
- f. **Response to tailoring instructions.** When data are delivered in the form of a traditional document, paragraphs that have been tailored out of the DID shall result in the corresponding paragraph number and title in the document, followed by "This paragraph has been tailored out." For data delivered in an alternative form, this representation need occur only in the table of contents or equivalent.
- g. **Multiple paragraphs and subparagraphs.** Any section, paragraph, or subparagraph in this DID may be written as multiple paragraphs or subparagraphs to enhance readability

7.2 **Content.** Content requirements begin on the following page.

The numbers shown designate the paragraph numbers to be used in the document. Each such number is understood to have the prefix "7.2" within this DID. For example, the paragraph numbered 1.1 is understood to be paragraph 7.2.1.1 within this DID.

1. **Scope.** This section shall be divided into the following paragraphs.
 - 1.1 **Identification.** This paragraph shall contain a full identification of the system and the CSCIs to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s).
 - 1.2 **System overview.** This paragraph shall briefly state the purpose of the system and the software to which this document applies. It shall describe the general nature of the system and software; summarize the history of system development, operation, and maintenance; identify the project sponsor, user, developer, and support agencies; identify current and planned operating sites; and list other relevant documents.
 - 1.3 **Document overview.** This paragraph shall summarize the purpose and contents of this document.
2. **Referenced documents.** This section shall list by document number and title all documents referenced in this document. This section shall also identify the source for all documents not available through normal Government stocking activities.
3. **Version description.** This section shall be divided into the following paragraphs to identify and describe a software version containing one or more Computer Software Configuration Items (CSCIs).
 - 3.1 **Inventory of materials released.** This paragraph shall list:
 - a. All physical media and documentation that make up the new version
 - b. All operation/support documents that are not a part of the delivered package, but are required to operate, load, or regenerate the software version
 - c. Security and privacy considerations for these items
 - d. Required safeguards in handling the materials
 - e. Duplication instructions and restrictions

- 3.2 Inventory of CSCI contents.** This paragraph shall identify all software that is part of the software version. If source code is included, it shall be identified in the same sequence as is used to organize the source code listings for delivery. Security and privacy considerations shall be included, as applicable, including the classification levels of the version contents.
- 3.3 Changes installed.** This paragraph shall contain a list of all changes incorporated into the software version since the previous version, with a cross reference to the affected system and software specifications. If change classifications have been used, such as the Class I/Class II changes described in MIL-STD-973, the changes shall be separated into these classifications. This paragraph shall also identify, as applicable, the problem reports, Engineering Change Proposals (ECPs), and Specification Change Notices(s) (SCNs) associated with each change. Note: This paragraph does not apply to the initial software version.
- 3.4 Adaptation data.** This paragraph shall identify or reference all unique-to-site data contained in the software version. For software versions after the first, this paragraph shall describe changes made to the adaptation data.
- 3.5 Interface compatibility.** This paragraph shall indicate other systems and configuration items, including version numbers, with which this software version interfaces and any impacts on those interfaces due to the changes incorporated in this version. Note: This paragraph does not apply to the initial software version.
- 3.6 Bibliography of reference documents.** For the initial software version, this paragraph shall list all documents pertinent to the CSCI(s) in the version. For subsequent versions, this paragraph shall identify changes to the listed documents.
- 3.7 Operational effect of changes.** This paragraph shall contain a subparagraph describing the operational effect of each change listed in 3.3 above. This description shall include problems corrected, new or changed capabilities, and other operational effects.

3.8 Installation instructions. This paragraph shall provide, as applicable:

- a. Instructions (either directly or by reference) for installing the software version
- b. Identification of other changes that have to be installed for this version to be used
- c. Any security, privacy, or safety precautions relevant to the installation
- d. Test procedures for determining whether the version has been installed properly
- e. A telephone number or access to a "help desk" to call if there are problems

3.9 Possible problems and known errors. This paragraph shall identify any possible problems or known errors with the software version, any steps being taken to resolve the problems or errors, and instructions (either directly or by reference) for recognizing, avoiding, correcting, or otherwise handling each one.

4. Software product specification. This section shall be divided into the following paragraphs to provide source code listings, compilation/build procedures needed to convert the source code into executable code, and (by inclusion or reference) design information needed to understand and modify/enhance the source code as required.

4.1 Software design. This paragraph shall contain, or reference an appendix or other deliverable document that contains information describing the design of one or more "as-built" (delivered) CSCI(s). The information shall be the same as that required in a Software Design Document (SDD). If this document or its equivalent is to be delivered for the "as-built" CSCI, this paragraph shall reference it. If not, the information shall be provided in this document. Information provided in headers, comments, and code of the source code listings may be referenced and need not be repeated in this paragraph. If the SDD is included in an appendix, the paragraph numbers and page numbers need not be changed.

4.2 Source code listings. This paragraph shall contain, or reference an appendix that contains, the source code listings for one or more CSCIs. This paragraph shall

provide an index that cross-references each software unit to the location in the listings where the code corresponding to that unit can be found and any comments useful in accessing or interpreting the listings.

- 4.3 Compilation/build procedures.** This paragraph shall describe, or reference an appendix that describes, the compilation/build process used to create the executable software product from the source code. It shall specify the compiler(s)/assembler(s) used; other hardware and software needed; any settings/options/conventions used; and procedures for compiling/ assembling, linking, and building the CSCI and the software system/subsystem containing the CSCI, including variations for different sites, configurations, versions, etc.
- 4.4 Measured resource utilization.** This paragraph shall specify the measured resource utilization of the CSCI(s) at the time of delivery.
- 4.5 Modification procedures.** This paragraph shall describe procedures that must be followed to modify the CSCI(s). It shall include or reference information on the following, as applicable:
- a. Support facilities, equipment, and software, and procedures for their use
 - b. Databases/data banks used by the CSCI and procedures for using them
 - c. Design or coding conventions to be followed
 - d. Compilation/build procedures if different from those above
 - e. Integration and testing procedures to be followed
 - f. Error conditions, and procedures for responding to them
- 5. Notes.** This section shall contain any general information that aids in understanding this document (e.g., background information, glossary). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.
- A. Appendixes.** Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each

appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, etc.).

| | |
|--|--|
| COAST GUARD DATA ITEM DESCRIPTION | APPROVAL DATE: |
| <p>Public reporting burden for collection of this information is estimated to average 110 hours per response, including the time for reviewing instructions, searching for existing data sources, gathering and maintaining the data needed and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to COMMANDANT (G-SCC-2), 2100 Second Street S.W., Room 6410, Washington, DC 20593-0001.</p> | |
| 1. TITLE: Coast Guard Progress Report (CG-PR) | 2. IDENTIFICATION NUMBER: DI-CGSDDS-008 |
| 3. DESCRIPTION/PURPOSE: 3.1 The Progress Report provides a means for the Contractor to indicate to the Government the progress of work and the status of the program and of the assigned tasks, to report costs, and to inform of existing or potential problem areas. | |
| CLIN NUMBER(s): | 5. OFFICE OF PRIMARY RESPONSIBILITY: USCG Software Policy & Standards Division (G-SCC-2) |
| 6. APPLICATION/INTERRELATIONSHIP: 6.1 The Progress Report is specific to a Delivery Order. Upon approval from the Government Contracting Officer, the Contractor may consolidate the Progress Reports for Delivery Orders for a specific project. | |
| 7. PREPARATION INSTRUCTIONS: 7.1 <u>General Instructions.</u> The document shall be developed according to the following format specifications. This report shall be typewritten on standard size (e.g., 8 1/2" by 11") white paper and securely stapled. Pages shall be sequentially number. All attachments shall be identified and referenced in the text of the report. The report shall be prepared in the contractor's format and shall be legible and suitable for reproduction. <p style="text-align: right;">(Continued on Page 2)</p> | |
| 8. DISTRIBUTION STATEMENT: <u>DISTRIBUTION STATEMENT A.</u> This DID will be incorporated into a COMDTINST along with the approved CG Software Development & Documentation Standards (CG-SDDS). | |

7. PREPARATION INSTRUCTIONS (continued)

7.2. Content. The report shall contain at a minimum the following information.

- a. The Point of Contact for the Contractor
- b. The Contract Number
- c. The Delivery Order Number
- d. The Reporting Period
- e. Date Prepared
- f. Preparer's Name and Phone Number
- g. Tasks/deliverables completed during reporting period
- h. Status of tasks/deliverables in progress
- i. Problems encountered during the reporting period
- j. Summary of important meetings, briefings, trips and conferences during the reporting period
- k. Results obtained related to previously-identified problem areas, with conclusions and recommendations
- l. Tasks scheduled to be performed in the next reporting period
- m. Anticipated difficulties during the next reporting period
- n. Meetings, briefings, trips and conferences scheduled during the next reporting period
- o. Action items requiring Government resolution
- p. Status of schedule
- q. Any significant changes to the contractor's organization or method of operation to the project management network

- r. Requirements for changes in key personnel for whom Personnel Data Forms have been submitted
- s. Cost incurred during reporting period and total delivery order expenditures as of reporting date; Cost-Type contracts must also include the following:
 - 1. Projection of the entire budget by task, revised in each report as necessary with written explanation for each change
 - 2. Staff hours by labor category versus staff hours planned by task; breakdown shall represent the quarter of the month being reported on as well as the contract through the quarter
 - 3. Total cost by cost element versus budgeted task; breakdown shall represent the quarter of the month being reported on as well as the contract through the quarter
 - 4. Statement that the contract can or cannot be completed within the total estimated cost
 - 5. Statement of budget projection for the next reporting period
 - 6. A statement of variance analysis

| | |
|---|--|
| COAST GUARD DATA ITEM DESCRIPTION | APPROVAL DATE: |
| Public reporting burden for collection of this information is estimated to average 110 hours per response, including the time for reviewing instructions, searching for existing data sources, gathering and maintaining the data needed and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to COMMANDANT (G-SCC-2), 2100 Second Street S.W., Room 6410, Washington, DC 20593-0001. | |
| 1. TITLE: Coast Guard Consolidated Software Development Document (CG-CSDD). | 2. IDENTIFICATION NUMBER: DI-CGSDDS-007 |
| 3. DESCRIPTION/PURPOSE: 3.1 The Consolidated Software Development Document (CG-CSDD) is a single document describing all aspects of a software development project. It contains, in summary form, the contents of all other DIDs required by the Coast Guard Software Development and Documentation Standards (CG-SDDS). A CG-CSDD is written as an alternative to the individual DIDs listed in the CG-SDDS. It is best suited to a small project on which a single document can represent all relevant information. | |
| CLIN NUMBER(s): | 5. OFFICE OF PRIMARY RESPONSIBILITY: USCG Software Policy & Standards Division (G-SCC-2) |
| 6. APPLICATION/INTERRELATIONSHIP: 6.1 This Data Item Description (DID) contains the format and contents preparation instructions for data generated under the work task described in paragraphs 5.1.1, 5.1.2, 5.1.3, 5.1.4, 5.1.5, 5.3.2, 5.3.3, 5.4.1, 5.4.2, 5.5.1, 5.5.2, 5.6.1, 5.6.2, 5.6.3, 5.7.1, 5.7.2, 5.7.3, 5.10.1, 5.10.2, 5.10.6, 5.11.1, 5.11.2, 5.11.6, 5.12.1, 5.12.2, 5.12.6, 5.13.6, 5.14.1, 5.14.2, 5.14.3, 5.14.4, 5.14.5, 5.15.3, 5.15.4, 5.21.1, 5.21.2 and 5.21.3 of CG-SDDS (COMDTINST 5234.4). 6.2 Commercial, or other existing documents may be substituted for all or part of this document if they contain the required data. | |
| 7. PREPARATION INSTRUCTIONS: 7.1 <u>General Instructions.</u> The document shall be developed according to the following format specifications: <ul style="list-style-type: none"> a. <u>Automated Techniques.</u> Use of automated techniques is encouraged. The Contract Data Requirements List (CDRL) should specify whether the data are to be delivered on paper or electronic media; any requirements on the electronic representation (such as ASCII, CALS, or compatibility with a specific word processor or other software); whether the data may be delivered in contractor format rather than in the format specified herein; and whether the data may reside in a computer-aided software engineering (CASE) or other automated tool rather than in the form of a traditional document. b. <u>Alternate presentation styles.</u> Charts, tables, matrices, and other presentation styles are acceptable substitutes for text when data required by this DID can be made more readable using these styles. <p style="text-align: right;">(Continued on Page 2)</p> | |
| 8. DISTRIBUTION STATEMENT: <u>DISTRIBUTION STATEMENT A.</u> This DID will be incorporated into a COMDTINST along with the approved CG Software Development & Documentation Standards (CG-SDDS). | |

Exhibit HH Coast Guard Data Item Description

7. PREPARATION INSTRUCTIONS (continued)

- c. **Title page or identifier.** When data are delivered in the form of a traditional document, the document shall include a title page containing, as applicable: document number; volume number; version/revision indicator; security markings or other restrictions on the handling of the document; date; document title; name, abbreviation, and any other identifier for the system, subsystem or item to which the document applies; contract number; CDRL item number; organization for which the document has been prepared; name and address of preparing organization. For data delivered in an alternative form, this information shall be included on external and internal labels or by equivalent identification methods.
 - d. **Table of contents.** When data are delivered in the form of a traditional document, the document shall contain a table of contents providing the number, title, and page number of each titled paragraph, figure, table, and appendix. For data delivered in an alternative form, this information shall consist of an internal or external table of contents containing pointers to, or instructions for accessing, each paragraph, figure, table, and appendix or their equivalents.
 - e. **Page numbering/labeling.** When data are delivered in the form of a traditional document, each page shall contain a unique page number and display the document number, including version, volume, and date, as applicable. For data delivered in an alternative form, files, screens, or other entities shall be assigned names or numbers in such a way that desired data can be indexed and accessed.
 - f. **Response to tailoring instructions.** When data are delivered in the form of a traditional document, paragraphs that have been tailored out of the DID shall result in the corresponding paragraph number and title in the document, followed by "This paragraph has been tailored out." For data delivered in an alternative form, this representation need occur only in the table of contents or equivalent.
 - g. **Multiple paragraphs and subparagraphs.** Any section, paragraph, or subparagraph in this DID may be written as multiple paragraphs or subparagraphs to enhance readability
- 7.2 **Content.** Content requirements begin on the following page. The numbers shown designate the paragraph numbers to be used in the document. Each such number is understood to have the prefix "7.2" within this DID. For example, the paragraph

numbered 1.1 is understood to be paragraph 7.2.1.1 within
this DID.

1. **Scope.** This section shall be divided into the following paragraphs.
 - 1.1 **Identification.** This paragraph shall contain a full identification of the system and the CSCIs to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s).
 - 1.2 **System overview.** This paragraph shall briefly state the purpose of the system and the software to which this document applies. It shall describe the general nature of the system and software; summarize the history of system development, operation, and maintenance; identify the project sponsor, user, developer, and support agencies; identify current and planned operating sites; and list other relevant documents.
 - 1.3 **Document overview.** This paragraph shall summarize the purpose and contents of this document.
 - 1.4 **All manuals** (e.g., Software User Manual) described in this DID shall be delivered as separately bound Appendices.
2. **Referenced documents.** This section shall list by document number and title all documents referenced in this document. This section shall also identify the source for all documents not available through normal Government stocking activities.
3. **Plans.**
 - 3.1 **Software development plan.** This paragraph shall be divided into subparagraphs as needed to describe the contractor's plans for developing the software. Included shall be:
 - a. An overview of requirements and constraints defining/affecting the work to be done
 - b. Plans for complying with CG-SDDS general requirements, including as applicable, the software development process to be used; general software engineering plans (methods to be used and approach for reusable software, safety analysis, meeting processing resource and reserve requirements, and recording rationale), and general software testing plans (approach for achieving

the required level of independence, testing on the target computer system or an equivalent, and stressing the software)

- c. Plans for complying with CG-SDDS detailed requirements, including as applicable, the approach to be followed in each build for developing project plans; establishing and maintaining a software development environment; performing or participating in system requirements analysis; performing or participating in system design analysis; software requirements analysis; software architectural design; software detailed design; coding and unit testing; unit integration and testing; CSCI testing; CSCI integration and testing; performing or participating in system testing; preparing for software use and support; preparing for software delivery; software product evaluations; software configuration management; joint (customer/contractor) reviews; software development management; and any other software development activities
- d. Schedule(s) and activity network(s) for the project
- e. Project organization and resources, including organizational structure to be used; personnel resources (staff-loading, skill levels, locations, clearances); contractor facilities to be used; Government furnished items required and dates needed; other required resources, plans for obtaining them, and need/availability dates
- f. Software Quality Assurance. Subparagraphs to describe the approach to be followed for software quality assurance. The planning in each subparagraph shall cover all requirements described in paragraphs 5.21, 5.21.1, 5.21.2, and 5.21.3 of the CG-SDDS.

3.2 Software installation plan. This paragraph shall be divided into subparagraphs as needed to describe the contractor's plans for installing the software at user sites. These plans shall include, as applicable:

- a. Installation overview: list of sites; installation schedule; point of contact for questions; required materials; overview of briefings and training to be given; overview of tasks to be performed and responsible

organizations; description of required personnel;
overview of security considerations

- b. Information needed by computer operations personnel at each site: schedule of activities; inventory of required software, facilities, and accommodations; composition of the installation team; each member's tasks; procedures for installation, conversion from the old system, and data updates during the installation
- c. Information needed by users at each site: schedule of activities; procedures for the installation, conversion from the current system, and data updates during installation

3.3 Software support plan. This paragraph shall be divided into subparagraphs as needed to describe the contractor's plans for transitioning the software to the support agency. These plans shall describe, as applicable:

- a. Resources required to support the deliverable software: facilities; hardware; software; data; documentation; personnel
- b. Procedures that contractor may wish to recommend to the support agency for supporting the deliverable software
- c. Plans for training personnel to manage and implement support of the deliverable software: schedule; location; types of training
- d. Anticipated areas of change to the deliverable software
- e. Plans for transitioning the deliverable software to the support agency: activities; roles and responsibilities; resources required; source of each resource; schedules; procedures for installation and checkout in the support environment

4. Concept and requirements.

4.1 Operational concept. This paragraph shall be divided into subparagraphs as needed to describe the operational concept for the system. This concept shall describe, as applicable:

- a. The current system or situation, including background,

objectives, and scope; operational policies/constraints; description of the current system/situation (operational environment; major components and the interconnections among them; interfaces to external systems or procedures; capabilities/functions; inputs, outputs, data flow, and manual and automated processes; performance characteristics; quality attributes; and provisions for safety, security, privacy, and continuity of operations); variations in different states or modes of operation; users or personnel involved; support/ maintenance concept and environment

- b. Justification for and nature of changes, including: new/modified factors that require a change; deficiencies in the current system or situation; new or modified capabilities/functions or other changes needed; priorities among the changes; changes considered but not included; and assumptions and constraints that apply
- c. The concept for a new or modified system, including all items listed in part a.
- d. Operational scenarios that illustrate the new or modified system
- e. Summary of impacts on the user, development, and support agencies
- f. Summary of improvements, disadvantages/limitations, alternatives and trade-offs

4.2 System/segment requirements. This paragraph shall be divided into subparagraphs as needed to specify the requirements for a system (or segment). This paragraph shall include:

- a. Identification of any states or modes in which the system is required to operate
- b. Itemized requirements, sufficient for acceptance, regarding each of the following, as applicable, annotated to identify variations in different states and modes:
 - 1) System capabilities (or functions/subjects/objects) and associated parameters such as response times, accuracy, etc.
 - 2) Databases/data banks that must be incorporated into the system

- 3) Interfaces with other systems, equipment, software, databases, and users
 - 4) Physical characteristics, such as weight limits, dimensional limits, color
 - 5) System quality factors, such as reliability, maintainability, availability
 - 6) The environment within which the system is required to operate
 - 7) Design and construction standards, including safety, security, and privacy
 - 8) System documentation; logistics considerations; personnel and training
- c. The order of precedence or importance of the requirements
 - d. Qualification methods to be used to verify compliance with each requirement
 - e. Requirements for delivery, including packaging and handling

4.3 CSCI requirements. This paragraph shall be divided into subparagraphs as needed to specify the requirements for each CSCI. It shall include, as applicable:

- a. Itemized requirements for the CSCI, sufficient for acceptance, regarding each of the following, as applicable, annotated to identify variations in different states and modes:
 - 1) CSCI-external interfaces (4.4 may be referenced)
 - 2) CSCI capabilities (or functions/subjects/objects) and associated parameters such as response times, accuracy, etc.
 - 3) CSCI-internal interfaces and internal data elements
 - 4) Unique-to-site data, conditions, or parameters
 - 5) Sizing and timing; safety, security, and privacy
 - 6) Design constraints; software quality factors allocated to the CSCI
 - 7) Considerations regarding human performance/human engineering
- b. A two-way mapping between CSCI requirements and the system requirements allocated to the CSCI (see 5.1.c)
- c. The qualification methods to be used to verify compliance with each requirement.
- d. Requirements for delivery of the CSCI, including

packaging and handling.

4.4 Interface requirements. This paragraph shall be divided into subparagraphs as needed to specify requirements for one or more interfaces between one or more CSCIs and other configuration items or critical items. These requirements may describe the interface characteristics of existing items or specify the required interface characteristics of new or to-be-modified items. This paragraph shall include, as applicable:

- a. Identification and overview of each interface to which this paragraph applies
- b. The requirements for each interface:
 - 1) Requirements pertaining to transmitted data elements, including, as applicable: identifier; description; source(s); recipient(s); units of measure; limit/range of values; accuracy; precision/resolution; timing characteristics; legality checks; data type; representation/format; sequence, priorities, or other dependencies
 - 2) Identification of each message or other data assembly that must be transmitted between the interfacing items, and required assignment of data elements to each
 - 3) The required priority among the interfaces, data elements, messages, or assemblies transmitted between the interfacing items
 - 4) Specification of each interface/communications protocol that must be used, including as applicable: fragmentation/reassembly of messages; message formatting; legality checks, error control and recovery procedures; synchronization; flow control; data transfer rate; routing, addressing, and naming conventions; transmission services; status and other reporting features; security and privacy features
- c. The qualification methods to be used to verify compliance with each requirement

5. Design.

5.1 System/segment design. This paragraph be divided into subparagraphs as needed to describe the design of a system (or segment), including:

- a. System behavioral design: decisions that transcend system internal structure, such as: allowed or expected inputs; planned behavior in response to each input; planned outputs; decisions on system databases/data banks; type of flexibility to be built into the system; levels of availability, integrity, and confidentiality to be offered
- b. System architecture, including:
 - 1) Internal structure of the system: segments, HWCIs, CSCIs
 - 2) Relationships among the segments, HWCIs, and CSCIs
 - 3) Identification, purpose, and description of each external interface of the system
- c. Identification of system requirements allocated to each HWCI, CSCI, and system-internal interface
- d. Description of processing resources to be used
- e. Traceability from system requirements to HWCIs, CSCIs, manual operations

5.2 CSCI design. This paragraph shall be divided into subparagraphs as needed to describe the design of one or more CSCIs. The description of each CSCI shall include:

- a. CSCI-wide behavioral design: decisions that transcend CSCI internal structure
- b. Rules/schemes/conventions used in designing the CSCI and expressing the design.
- c. CSCI architectural design, indicating the role of the CSCI within the system architecture; the internal architecture of the CSCI (including name/ID and purpose of each software unit; and CSCI structure, including flow of execution/data among units); any non-developmental software to be incorporated into the CSCI; and allocation of memory and processing time to the software units
- d. CSCI detailed design, describing for each software unit:

- 1) Purpose
 - 2) Requirements allocated to the software unit
 - 3) Derived design requirements (e.g., algorithms used)
 - 4) Constraints, limitations, or unusual features
 - 5) Programming language and rationale, if different from the CSCI language
 - 6) Reusable software to be used
 - 7) Program library in which the software unit is to be placed
 - 8) Inputs, outputs, and other data elements of the software unit, including, as applicable for each: identifier; description; units of measure; limit/range of values; accuracy; precision/resolution; priority; frequency of input, calculation, or output; legality checks to be performed; data type; data representation, format, or structure; sources, including other software units where the data element is set; destinations, including other software units where used; maximum size and storage needs; and access method, such as random or sequential
 - 9) The logic flow to be used by the software unit, including conditions under which software unit execution is initiated; conditions under which control is passed to other software units; response and response time to each input; and dynamically controlled sequencing during the software unit's operations
- e. Traceability from the CSCI requirements to the software units

5.3 Interface design. This paragraph shall be divided into subparagraphs as needed to describe the specific interface characteristics selected to respond to the CSCI-external interface requirements. It shall include, as applicable:

- a. Identification and overview of each CSCI-external interface to which this paragraph applies, including interface diagrams as appropriate, and a characterization of each interface (sequential/concurrent operation, real-time, store-and-retrieve, etc.)

- b. The following information for each interface, as applicable:
- 1) Description of each data element to be transmitted, including as applicable: identifier; description; source(s); recipient(s); units of measure; limit/range of values; accuracy; precision/resolution; timing characteristics; legality checks; data type; representation/format; sequence or other dependencies
 - 2) Identification of each message or other data assemblies transmitted between the interfacing items, and assignment of data elements to each message or assembly
 - 3) The relative priority of the interface and of each data element, message, or assembly transmitted between the interfacing items
 - 4) Description of each interface/communications protocol to be followed, including as applicable: fragmentation/reassembly of messages; message formatting; legality checks, error control and recovery procedures; synchronization; flow control; data transfer rate; routing/addressing/naming conventions; transmission services; status and other reporting features; security and privacy features

5.4 Database design. This paragraph shall be divided into subparagraphs as needed to describe the design of one or more databases. It shall include, as applicable, for each database:

- a. An overview of the database, including:
- 1) Database environment: systems that will use the database; relationship to other databases; estimated storage requirements; overview of components and files; data communications environment
 - 2) Internal labeling conventions used in the database design
 - 3) Overview of conceptual and physical organization and design considerations
 - 4) Instructions for personnel who will contribute to or use the database
 - 5) Support software available for handling the database
 - 6) Overview of security, privacy and criticality considerations

- b. Information needed to establish and administer the database, including:
 - 1) Identification of offices responsible for database administrative functions
 - 2) Identification of the Database Management System (DBMS), hardware configuration, utility software, and integrity and access controls
 - 3) Schema information, including: overall structure; rationale; data elements and the subschemas in which they are visible; description of each subschema; logical organization; physical structure; sizing formulas; recovery methods; cross-reference to requirements
 - 4) Area/file information, including: rationale; content of each area/file; description of each area/file; information about data storage; any limiting factors regarding allocation, expansion, paging, loading, size; recovery methods for reconstructing the necessary data and structure

- c. Information needed by applications developers to use the database, including:
 - 1) Detailed technical description of the database, including for each subschema or local view: name; purpose; a list and description of records; sets or predefined relations; areas/files; access routines and query path structures; and security/ privacy controls
 - 2) Information about any utility software to aid in database use
 - 3) Description of any error handling routines and procedures available
 - 4) List of all messages output during execution of database software

6. Software testing.

6.1 Software test planning. This paragraph shall be divided into subparagraphs as needed to describe plans for conducting CSCI, CSCI integration, and software system testing. Included, as applicable, shall be:

- a. A description of the software test environment at each test site, including hardware and software items; issues of security/privacy and Government rights; plans for installation, testing, and control of the environment; organizations that will participate in the testing; the number and skill types of personnel required during the test period; dates and time they will be needed; plans for orientation and training; and identification of tests to be performed at the site (by reference to tests in part c.)
- b. General information about the testing to be performed, including levels of tests to be performed (e.g. CSCI, CSCI integration, system); classes of tests (e.g, stress tests, timing tests); any test requirements that apply to all, or a group of, tests; extent of testing to be performed (e.g., specified sampling of possible inputs) and rationale for the extent selected; planned sequence or progression of tests; and data recording, reduction, and analysis procedures to be used during and after the tests
- c. Identification of planned tests for each item to be tested, including test name; objective; test level (using the levels defined above); test class (using the classes defined above); qualification method(s) specified in the requirements specification(s); cross reference to the software requirements addressed by this test; special test requirements; type of data to be recorded; type of data recording/reduction/analysis to be employed; assumptions and constraints; and security and privacy considerations
- d. Schedules for conducting the tests identified above, including pretest activities; preparation of test inputs and databases; any orientation or training associated with the testing; the testing itself; and any period for review/approval of test results

6.2 Software test descriptions. This paragraph shall be divided into subparagraphs as needed to describe the test cases and test procedures to be used for CSCI, CSCI integration, and software system testing. For each item to be tested, this paragraph shall include:

- a. Preparations for each test, including security/privacy considerations; test schedule; and pre-test procedures for hardware, software, and other aspects of the test
- b. Description of each test, including test name/identifier; security and privacy considerations; and description of each test case to be included in the test:
 - 1) Cross-reference to the software requirements addressed by the test case
 - 2) Means of control of test sequence (manual, semi-automatic, or automatic)
 - 3) Prerequisite conditions that must be established
 - 4) Test inputs: description, source, timing/sequence, means of control
 - 5) Expected results, both intermediate and final, as applicable
 - 6) Criteria for evaluating the results
 - 7) Test procedure: numbered steps to be followed in initiating, carrying out, and analyzing the results of the test case, including, as applicable, alternative actions, expected results, and evaluation criteria for each step
 - 8) Any assumptions or constraints applicable to the test case

6.3 Software test report. This section shall be divided into subparagraphs as needed to describe the results of CSCI, CSCI integration, and software system testing. For each item tested, this paragraph shall include:

- a. An overview of the results of each test, including reference to a test log; description of the test's completion status (for example, "all results as expected," "problems encountered," "deviations required"); and, when not "as expected," a summary of the problems or deviations that occurred
- b. Detailed results of each test, including the completion status of each test case associated with the test; a description of each problem encountered, including the test procedure step where the problem was encountered and reference to associated problem report(s); and a description of each deviation from the documented test cases/test procedures, including test procedure step

where the deviation occurred, nature of the deviation, rationale, and an assessment of its impact on the validity of the test case

- c. Evaluation and recommendations, including an overall assessment of each item under test; an assessment of the manner in which the test environment may be different from the operational environment and the effect of this difference on the capabilities tested; and any recommended improvements in the design, operation, or testing of the item(s) under test

7. User/operator manuals.

7.1 Software user manual. This paragraph shall be divided into subparagraphs as needed to tell a hands-on software user how to install, initiate, use, and terminate a single CSCI, a group of related CSCIs, or an overall software system. It include, as applicable:

- a. A software summary, describing software uses; functions performed; logical parts and communication paths from a user view; interfaces; performance characteristics; supervisory control; hardware and software that must be present for the software to run; contingencies and alternate modes of operation; and points of contact and procedures for obtaining assistance and report problems
- b. Access information, including information for first-time users (equipment familiarization; overview of access, security, and privacy features; and procedures the user must perform to be identified or authorized to access, install, or use the software); procedures for beginning work, including options and a possible problems; and description of how to cease or interrupt use
- c. A processing reference guide, including:
 - 1) An overview of software use
 - 2) Conventions such as use of colors, audible alarms, abbreviated vocabulary, etc.
 - 3) Detailed processing procedures, organized by function, menu, or other method, as appropriate, including options, examples, formats, messages, etc. as needed
 - 4) Description of any batch or background processing

performed by the software

- 5) Procedures for creating and retaining backup data
- 6) Procedures for restart/recovery and continuity of operations
- 7) List of error and other messages, their meaning, and action to be taken for each

7.2 Computer center software operator manual. This paragraph shall be divided into subparagraphs as needed to provide personnel in a computer center an operational description of a software system, its operating environment, and procedures for performing computer runs of the system. It shall include, as applicable:

- a. A non-technical description of the software, including intended uses; operations of the system and how they are interrelated; inventory of the software that makes up the system; list of files and databases/data banks referenced/created/updated by the system; list of reports produced by the system; overview of system processing, communications processes, and security, privacy, and continuity of operation
- b. A description of the runs to be performed, including:
 - 1) A list of the runs, purpose of each run, and software/jobs included in each
 - 2) Schedule of acceptable phasing of the system
 - 3) Setup and execution procedures for any software diagnostics
 - 4) Error messages and corresponding correction procedures
 - 5) Detailed information needed to execute runs, including control statements; run management information; files/databases/data banks input to or created/updated by the run; reports produced, including security/privacy, media, size, copies, recipients; instructions for reproducing and distributing reports; and procedures for restart/recovery and continuity of operations

7.3 Software input/output manual. This paragraph shall be divided into subparagraphs as needed to tell a user how to submit queries and other inputs to a batch or interactive software system that is run by computer center staff, and to describe the outputs that can be expected from the system.

It shall include, as applicable:

- a. A non-technical overview of the software, including: uses; capabilities; improvements; benefits; functions; logical parts; communication paths and techniques; interfaces; participating organizations; relationships to organizations that provide input or receive output; overview of the equipment, communications, and networks used by the system; relationships among logical parts of the system and the role of each; performance capabilities and constraints/limitations that can be expected; differences at times of emergencies and peacetime/wartime/conditions of alert; method used to store the data, including Database Management System (DBMS); and overview of inputs, processing, and outputs
- b. Instructions necessary to prepare inputs and interpret outputs, including procedures to initiate system operation; instructions and examples for preparing inputs; instructions and examples for interpreting output reports; use of the output by the operational area or activity that receives it; error codes generated by the software and appropriate actions to be taken; and procedures available to the user for testing/ diagnosing communications
- c. Instructions for preparation and processing of database/file queries, including preprogrammed query capabilities provided by the system; illustration of the data base/data bank format and content; instructions for preparing necessary query parameters; and instructions for sequencing runs and software for the queries
- d. Information on the use of terminals to accomplish processing, including capabilities for retrieval, display, and update of data; access procedures; procedures for display, update, and retrieval of data; recovery, error correction, and termination procedures

7.4 Computer system operator manual. This paragraph shall be divided into subparagraphs as needed to provide information for operating and performing diagnostics on one or more new or special-purpose computers. It shall include, as applicable:

- a. Name, identification number, version, and any other identifying information for the computer(s), peripheral

equipment, and other constituent parts

- b. Procedures for computer system preparation and shutdown, including procedures for power-on and power-off; procedures to initiate operation of the computer system; and shutdown procedures necessary to save files and terminate operation
- c. Procedures for operating the computer system, including input and output media and procedures to read/write on them; procedures to be followed for monitoring the system in operation; procedures to be followed for each trouble occurrence; and procedures to operate all off-line equipment/routines
- d. Diagnostic features, including a summary of diagnostic features; each diagnostic procedure to be followed; and available diagnostic tools and their application

8. Support information.

8.1 Version description. This paragraph shall be divided into subparagraphs as needed to provide describe a software version containing one or more Computer Software Configuration Items (CSCIs). It shall include, as applicable:

- a. An inventory of materials released, including software and documentation
- b. An inventory of all software that is part of the software version
- c. A list of all changes installed since the previous version, including change classifications, trouble reports, and change proposals, as applicable
- d. Identification of unique-to-site data contained in the software version
- e. Identification of interfacing items and the version's impact on interfaces with them
- f. List of documents pertinent to the version, or changes to them
- g. Operational effect of each change in the version
- h. Instructions for installing the version
- i. Identification of problems or known errors in the software version

8.2 Software product specification. This paragraph shall be divided into subparagraphs as needed to provide source code

listings, compilation/build procedures, and (by inclusion or reference) design information needed to understand and modify/enhance the source code as required. It shall include, as applicable:

- a. Information, or reference to information, describing the design of the as-built software
- b. Source code listings and an index and comments for accessing them
- c. Description of the compilation/build process for creating the executable software
- d. The measured resource utilization of the CSCI(s) at the time of delivery
- e. Procedures for modifying the CSCI(s)

9. **Notes**. This section shall contain any general information that aids in understanding this document (e.g., background information, glossary). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.

A. **Appendixes**. Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, etc.).

US Coast Guard
Software Development and Documentation Standards
(CG-SDDS COMDTINST 5234.4)
GUIDE
April 1996
US Coast Guard Headquarters
Software Policy & Standards Division (G-SCC-2)
2100 Second Street SW
Washington, DC 20593-0001

Overview

Purpose: The purpose of this guide to using the Coast Guard Software Development and Documentation Standards (CG-SDDS) is to provide an overview of applying the standard to a software development effort. The CG-SDDS is a tasking document that is supported by eight Data Item Descriptions (DIDs). The CG-SDDS supports a process-driven software development. A frequent complaint about software development under military standards is that it is document driven. Rather than focusing on the real work to be done, the developer must generate an endless series of documents, some of which have little to do with getting the job done. The contractor may have to translate real work products, such as data in computer-aided software engineering (CASE) tools, into paper documents or translate work products from the actual format in which work is being done to an artificial format imposed by Data Item Descriptions (DIDs).

DIDs as activity checklists. A unique feature of CG-SDDS is that it uses the DIDs to fully define many of the activities in the standard. The DIDs specify the information that is to be defined and recorded in carrying out the activities. This use of the DIDs applies regardless of whether the activities result in a tangible deliverable.

CG-SDDS requires work to take place, regardless of whether the results are made deliverable, and gives the Coast Guard access to the results in the developer's facility. When deliverables are needed, they are called for in the CDRL.

Many of the activities in CG-SDDS rely on DIDs to fully define them. For example, the activity of preparing the CG Software Plan (CG-SP) consists of defining and recording all information called for by the CG-SP DID. This requirement applies regardless of whether the CG-SP is in the form of a document and regardless of whether the resulting plan is deliverable. The DID is used as a checklist of information to be defined in carrying out the required activity. **This is a new role for DIDs.**

Figure 1 illustrates the role of DIDs and software products in CG-SDDS, emphasizing the distinction made between natural work products, the form in which they reside, and their status as deliverables or non-deliverables.

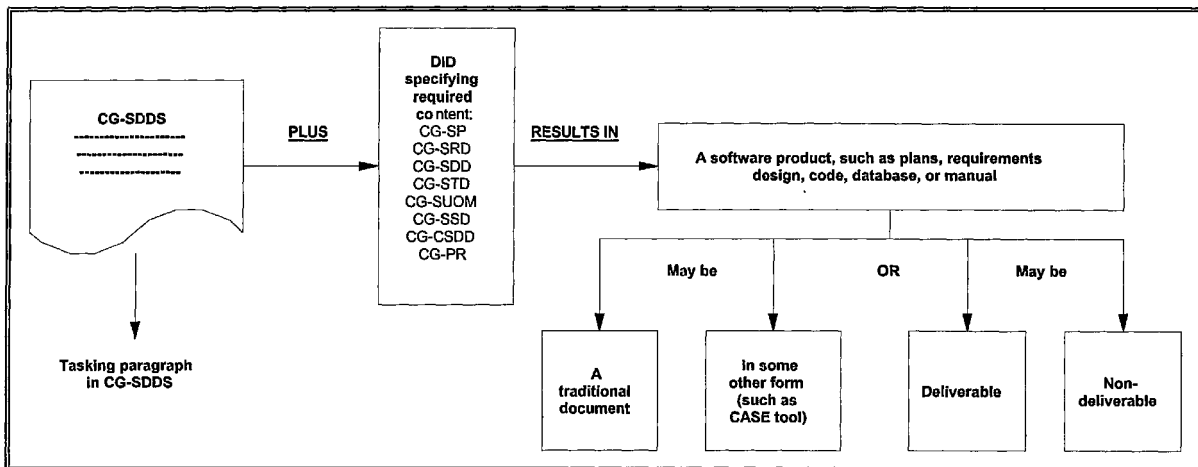


FIGURE 1: The role of DIDs and software products in CG-SDDS
FIGURE 1: The role of DIDs and software products in CG-SDDS.

Roles of plans in CG-SDDS. The basic philosophy of CG-SDDS is that the standard tells "what" the contractor must do. The developer proposes the "how" via the plans. The Coast Guard has the right to review and approve (or disapprove) the plans, and the contractor must adhere to the plans once approved. This philosophy makes the plans, especially the CG Software Plan (CG-SP), key documents in the project.

CG-SDDS philosophy regarding deliverables. CG-SDDS has been worded to differentiate between the planning and engineering activities that make up a software development project and the generation of deliverables. A key objective of this wording is to eliminate the notion that the Coast Guard must order a given deliverable in order to have planning or engineering work take place. Under the CG-SDDS, the planning and engineering work takes place regardless of which deliverables are ordered, unless a given activity is tailored out of the standard. In addition, joint technical reviews have been included to review the results of that work in its natural form, without the generation of deliverables. **Software products should be designated as deliverables only when there is a genuine need to have planning or engineering information transformed into a deliverable, recognizing that this transformation requires time and effort that would otherwise be spent on the engineering effort.**

Multiple formats for deliverables. Traditional deliverables take the form of paper documents exactly following DID formats. While this form works well for some deliverables, it is not the only form, and alternatives should be considered. One variation from paper documents is electronic word processing files containing these documents. This format saves paper, but still requires the contractor to format the information as required by the DID. Another variation is specifying that a paper or word processor document is to include all DID contents but may be in the developer's format. Yet another variation is allowing deliverables to take forms that are not traditional documents at all, such as data in computer-aided software engineering (CASE) tools. These variations can minimize the time spent transforming actual work products into deliverables.

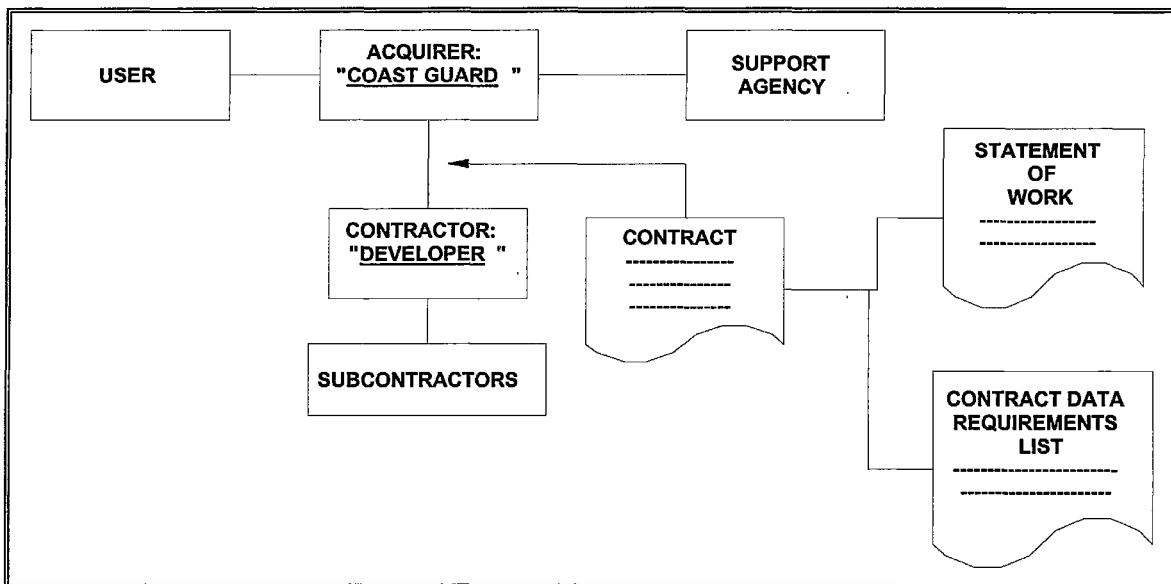


FIGURE 2: Players and agreements assumed by the CG-SDDS

CG-SDDS DIDs. Figure 3 provides a graphical depiction of the CG-SDDS and its supporting Data Item Descriptions (DIDs). Figure 4 provides a graphical depiction of the functional activities associated with applying the CG-SDDS and the supporting DIDs to a software development lifecycle. Following Figure 4 are two items: a one page step-by-step listing of key product reviews and completions of specific sections and documents (deliverables) generated by the CG-SDDS, and several pages with detailed descriptions of each process, document, and review shown in Figure 4. The detailed descriptions are listed in the order and lifecycle phase in which they occur.

Applying the CG-SDDS DIDs. Appendix A provides a sample statement of work (SOW) which contains language for applying the CG-SDDS and the supporting DIDs to a software development project.

FIGURE 3: Supporting Data Item Descriptions (DIDs)

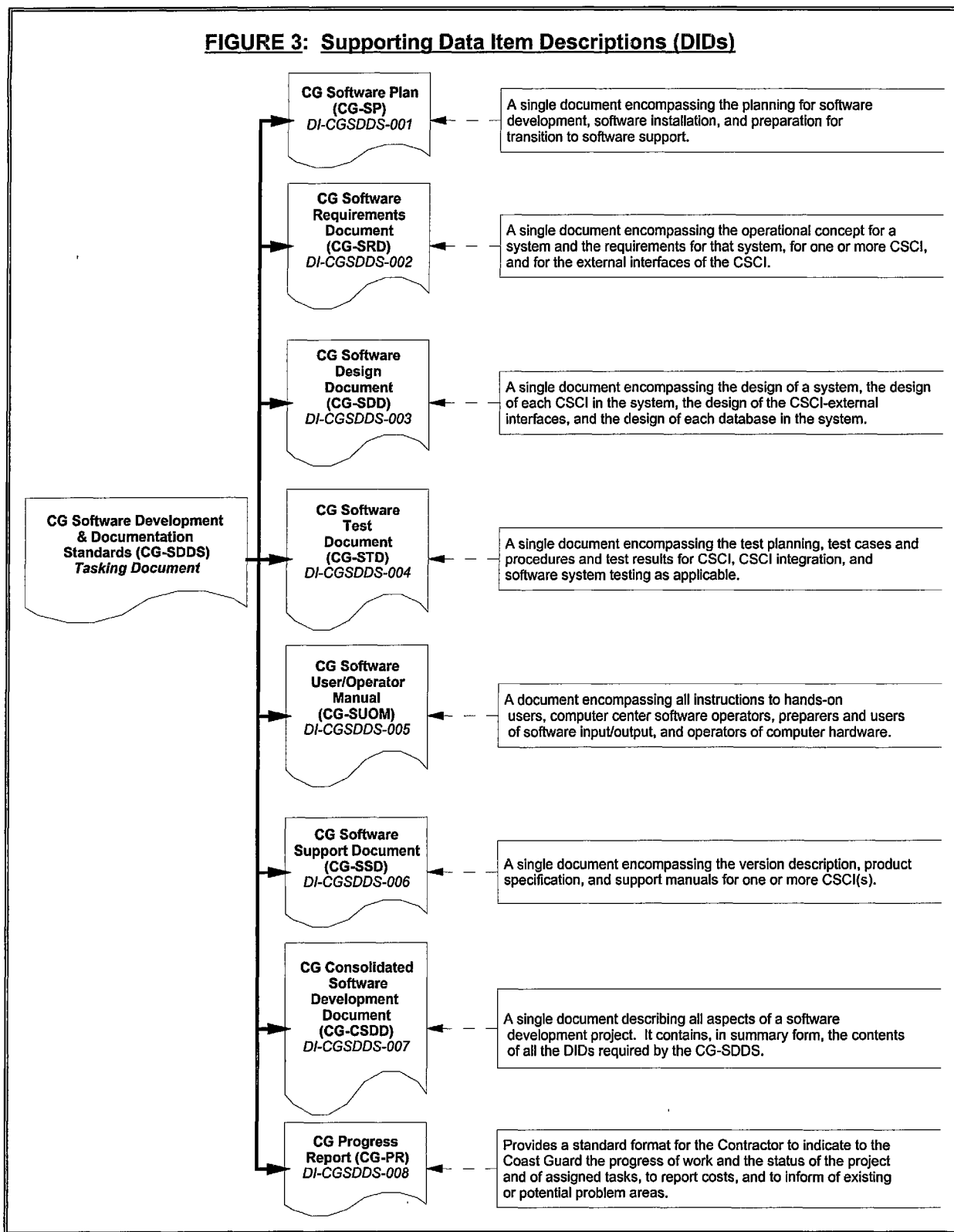


Figure 3: Supporting Data Item Descriptions

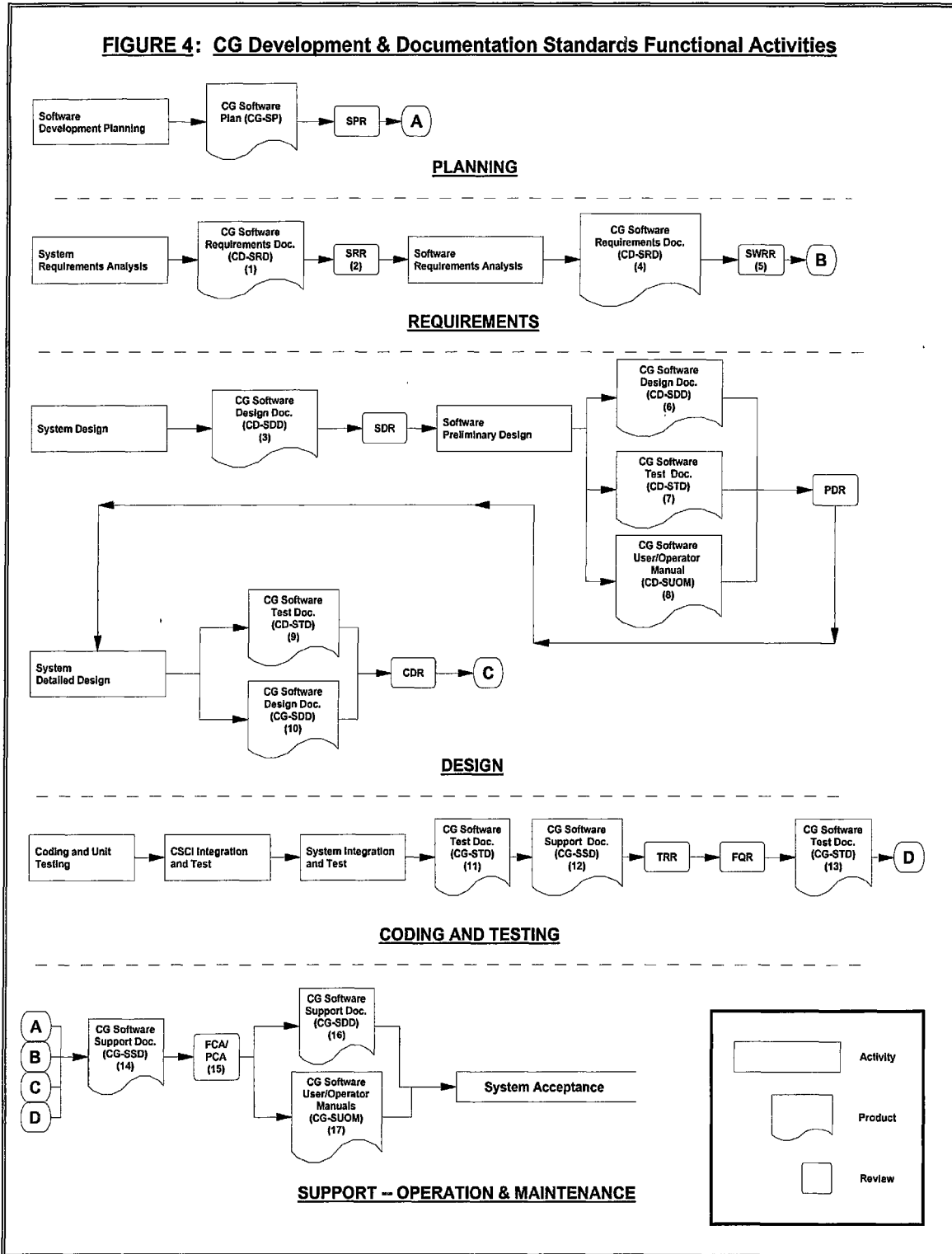


Figure 4: CG Development & Documentation Standards Functional Activities

CG Development & Documentation Standards Functional Activities (continued)

Refer to page 6, figure 4

- (1) **CG Software Requirement Doc. (CG-SRD)** Sections 1-4 (SSS) and preliminary Sections 5 and 6
- (2) **Software Requirements Review (SRR)** - Combine Operational Concepts/System Requirements Review. May also include Software Plan Review
- (3) **CG Software Design Doc. (CG-SDD)** - Sections 1 through 3
- (4) **CG Software Requirements Doc. (CG-SRD)** - Sections 5 and 6 Completed
- (5) **CG Software Requirements Doc. (CG-SRD)** - Completed
- (6) **CG Software Design Doc. (CG-SDD)** - Sections 5 and 6
- (7) **CG Software Test Doc. (CG-STD)** - Sections 1 through 3
- (8) **Preliminary CG Software User/Operator Manual (CG-SUOM)**
- (9) **CG Software Test Doc. (CG-STD)** - Section 4
- (10) Completed **CG Software Design Doc. (CG-SDD)** including Section 4 and updated section 5
- (11) **CG Software Test Doc. (CG-STD)** - Section 1 through 3 completed (with Procedures)
- (12) **CG Software Support Doc. (CG-SSD)** - Section 4 (Software Product Specification)
- (13) **CG Software Test Document (CG-STD)** - Software Test Report -(STR) completed CG-STD
- (14) **CG Software Support Document (CG-SSD)** - Section 4 (Version Description Documentation)
- (15) **May be held in lieu of, or in addition to, a Software Supportability Review and Software Usability Review**
- (16) **CG Software Support Document (CG-SSD)** - Completed
- (17) **CG Software User/Operator Manual (CG-SUOM)** -Completed

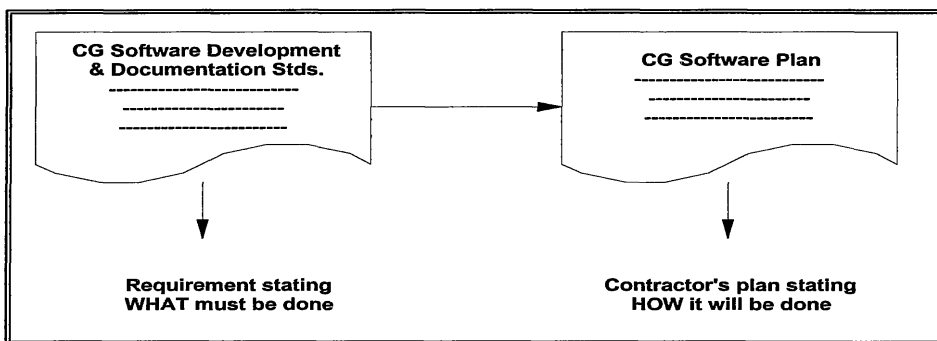
PLANNING PHASE:

Software Development Planning. Coast Guard developed requirements are provided to contractor(s). Based on the Coast Guard's requirements the contractor(s) develops a Software Plan that presents the contractor's understanding of the requirement, and the contractor's approach to fulfilling the requirement.

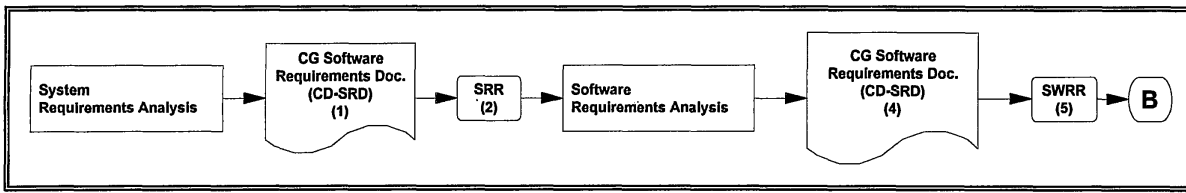
CG Software Plan (CG-SP). A single document encompassing the planning for software development, software installation, and preparation for transition to software support. Describes the contractor's plans for conducting a software development effort. Covers new development, modification, reuse, reengineering, maintenance, and all other activities resulting in software products. Provides the Coast Guard insight into, and a tool for monitoring, the processes to be followed for software development, the methods to be used, the approach to be followed for each activity, and project schedules, organization, and resources. Portions of the plan may be bound separately if the approach enhances their usability. Examples include plans for software configuration management and software quality assurance.

Software Plan Review (SPR). The SPR is the most important review of the software development effort. It serves as the foundation of understanding between the Coast Guard and the contractor(s) based on the contents of the contractor-developed Coast Guard Software Plan (CG-SP). The SPR will be conducted after the Coast Guard has received and reviewed the initial CG-SP from the contractor for the related software development initiative. A successful SPR is predicated upon the Coast Guard's determination that the *CG Software Plan (CG-SP)* forms a satisfactory basis for proceeding into System Requirements Analysis.

Methodology Independence. It is the CG-SDDS's goal to neither encourage nor preclude any particular software development methodology. As shown in the figure below, the standard lays out a set of activities that must be accomplished and defines a set of software products that must be produced, but leaves it to the developer to propose in the Coast Guard Software Plan (CG-SP) how those activities will be performed and how the products will be produced. **The Coast Guard has the opportunity to react to the methods in reviewing the (CG-SP).**



END PLANNING PHASE



REQUIREMENTS PHASE:

System Requirements Analysis. The analysis and documentation of system requirements that involves transforming an operational need into a system description, system performance parameters, and a system configuration through the use of an iterative process of analysis, design, trade-off studies, and prototyping.

CG Software Requirements Document (CG-SRD). A single document encompassing the operational concept for a system and the requirements for that system, for one or more Computer Software Configuration Items (CSCI), and for the external interfaces of the CSCIs. Specifies the requirements for a system and its supporting CSCI(s) and the methods to be used to ensure that each requirement has been met. The *CG-SRD* provides guidance for and comprises the following information:

Operation Capabilities Description, System/Segment Specification, Software Requirements Specification, and Interface Requirements Specification. (1) *CG-SRD* sections 1 through 4 and preliminary sections 5 through 6.

System Requirements Review (SRR). The object of this review is to ascertain the adequacy of the contractor's efforts in defining the system requirements via the Coast Guard Software Requirements Document (CG-SRD). The SRR will be conducted after the Coast Guard has received and reviewed the initial CG-SRR from the contractor for the related software development initiative. A successful SRR is predicated upon the Coast Guard's determination that the system related sections (Section 1 through 4) of the *CG Software Requirements Document (CG-SRD)* form a satisfactory basis for proceeding into preliminary software design. (2) *Combine Operation Concepts/Systems Requirement Review. May also include Software Plan Review (CG-SP).*

Software Requirements Analysis. Analyzes the software problem at hand and presents a complete specification of the desired external behavior of the software system to be built. The analysis involves partitioning system requirements into major subsystems and tasks, allocating those subsystems or tasks to software, and transforming these allocated system requirements into a description of software requirements and performance parameters through use of an iterative process of analysis, design, trade-off studies, and prototyping. Also called *software functional description software functional requirements, software requirements specification* by others.

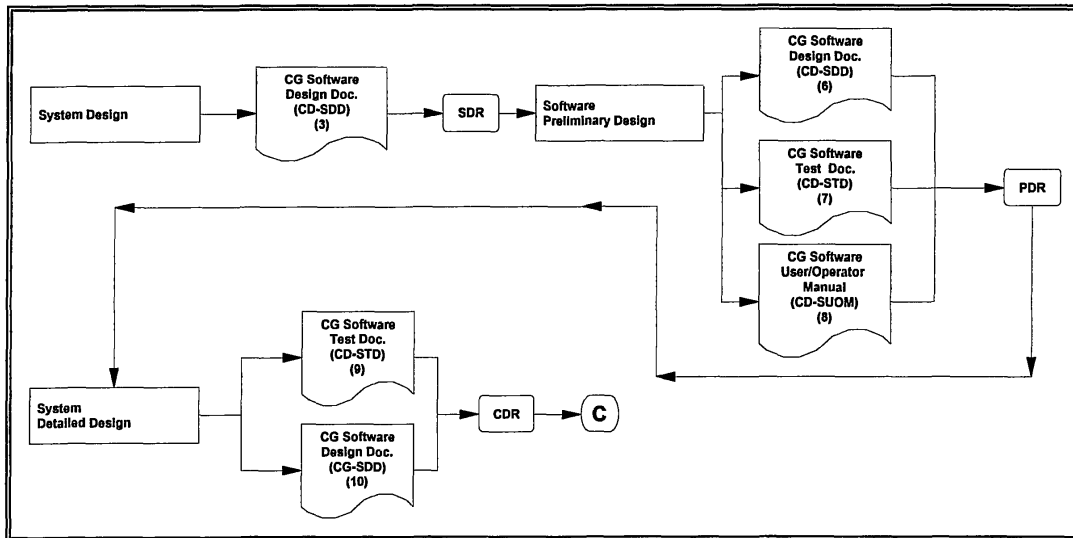
NOTE: The major difference between system requirements analysis and software requirements analysis is that the origin of system requirements lies in user needs while the origin of software requirements lies in the system requirements and/or specification.

CG Software Requirements Document (CG-SRD). A single document encompassing the operational concept for a system and the requirements for that system, for one or more Computer Software Configuration Items (CSCI), and for the external interfaces of the CSCIs. The *CG-SRD* specifies the requirements for a system and its supporting CSCI(s) and the methods to be used to ensure that each requirement will be met. The **CG-SRD** provides guidance for and comprises the following information:

Operation Capabilities Description, System/Segment Specification, Software Requirements Specification, and Interface Requirements Specification. (4) *CG-SRD* sections 5 and 6 completed.

Software Requirements Review (SWRR). A review of the finalized CSCI requirements and operational concept. The SWRR is conducted when CSCI requirements have been sufficiently defined to evaluate the contractor's responsiveness to and interpretation of the system, segment, or prime item level requirements. A successful SWRR is predicated upon the contracting agency's determination that the *CG Software Requirements Document (CG-SRD)* forms a satisfactory basis for proceeding into preliminary software design. (5) *CG-SRD* completed.

END REQUIREMENTS PHASE



DESIGN PHASE:

System Design. Analysis and determination of major system components and their purposes. Software and hardware requirements for each of the determined system components is separately specified. The process of defining the hardware and software architecture, components, modules, interfaces, and data for a system to satisfy specified system requirements.

CG Software Design Document (CG-SDD). A single document encompassing the design of a system, the design of each CSCI in the system, the design of the CSCI-external interfaces, and the design of each database in the system. Describes the system-wide design decisions, the system architectural design, and the detailed design needed to implement the software. Provides the Coast Guard visibility into the design and provides information needed for software support. (3) *CG-SDD sections 1 through 3.*

Software Design Review (SDR) This review shall be conducted to evaluate the optimization, correlation, completeness, and risks associated with the allocated technical requirements as presented in the Coast Guard Software Design Document (CG-SDD). The SDR will be conducted after the Coast Guard has received and reviewed the initial CG-SDD from the contractor for the related software development initiative. A successful SDR is predicated upon the Coast Guard's determination that sections 1 through 3 of the *CG Software Design Document (CG-SDD)* form a satisfactory basis for proceeding with the preliminary software design.

Software Preliminary Design. Decomposes the software system into its actual constituent (architectural) components and then iteratively decomposes those components into smaller and smaller subcomponents until the subcomponents located at the leaves of the resulting design tree are small enough for a person to be able to "get his or her arms around it" easily. Each of these modules is documented in terms of its inputs, outputs, and functions; also called *Preliminary Specifications, Preliminary High-level Design, Preliminary Architectural Design, and Preliminary Functional Design.*

DESIGN PHASE (continued)

CG Software Design Document (CG-SDD). A single document encompassing the design of a system, the design of each CSCI in the system, the design of the CSCI-external interfaces, and the design of each database in the system. Describes the system-wide design decisions, the system architectural design, and the detailed design needed to implement the software. Provides the Coast Guard visibility into the design and provides information needed for software support. (6) *CG-SDD sections 5 and 6*

CG Software Test Document (CG-STD). A single document encompassing the test planning, test cases and procedures, and test results for CSCI, CSCI integration, and software system testing as applicable. Provides a record of the qualification testing performed on a Computer Software Configuration Item (CSCI), a software subsystem, or other related item. Enables the Coast Guard to assess the testing and its results. The **CG-STD** provides guidance for and comprises the following information: a Software Test Plan, a Software Test Description, and a Software Test Report. (7) *CG-STD Sections 1 through 3.*

CG Software User/Operator Manual (CG-SUOM). A single document encompassing all instructions to hands-on users, computer center software operators, preparers and users of software input/output, and operators of computer hardware. The **CG-SUOM** provides guidance for and comprises the following information: a Software User Manual, a Computer Center Software Operator Manual, Software Input/Output Manual, Computer System Operator Manual. (8) *Preliminary CG Software User/Operator Manual.*

Preliminary Design Review (PDR) This review shall be conducted for each configuration item or aggregate of configuration items to:

- (1) evaluate the progress, technical adequacy, and risk resolution (on technical, cost, and schedule basis) of the selected design approach,
- (2) determine its compatibility with performance and engineering speciality requirements of the Hardware Configuration Item (HWCI) development specification,
- (3) evaluate the degree of definition and assess the technical risk associated with the selected development methods/processes, and
- (4) establish the existence and compatibility of the physical and functional interfaces among the configuration item and other items of equipment, facilities, computer software, and personnel.

For CSCIs, this review will focus on:

- (1) the evaluation of the progress, consistency, and technical adequacy of the selected top-level design and test approach,
- (2) compatibility between software requirements and preliminary design, and
- (3) on the preliminary version of the operation and support documents.

The PDR will be conducted after the Coast Guard has received delivery and reviewed the initial CG-SDD from the contractor for the related software development initiative. A successful PDR is predicated upon the Coast Guard's determination that sections:

- * 5 through 6 of the *CG Software Design Document (CG-SRD)*,
 - * 1 through 3 of the *CG Software Test Document (CG-STD)*, and
 - * the preliminary *CG Software User/Operator Manual (CG-SUOM)*,
- form a satisfactory basis for proceeding into System Detailed Design

DESIGN PHASE (continued)

System Detailed Design. Develop design descriptions of each software unit of the CSCI. Define and document algorithms for each module in the design tree that will be realized as code; also called *program design*.

CG Software Test Document (CG-STD). A single document encompassing the test planning, test cases and procedures, and test results for CSCI, CSCI integration, and software system testing as applicable. Provides a record of the qualification testing performed on a Computer Software Configuration Item (CSCI), a software subsystem, or other related item. Enables the Coast Guard to assess the testing and its results. The **CG-STD** provides guidance for and comprises the following information: a Software Test Plan, a Software Test Description, and a Software Test Report. (9) *CG-STD section 4.*

CG Software Design Document (CG-SDD). A single document encompassing the design of a system, the design of each CSCI in the system, the design of the CSCI-external interfaces, and the design of each database in the system. Describes the system-wide design decisions, the system architectural design, and the detailed design needed to implement the software. Provides the Coast Guard visibility into the design and provides information needed for software support. (10) *Completed CG Software Design Doc. including section 4 and an updated section 5.*

Critical Design Review (CDR). This review shall be conducted for each configuration item when detail design is essentially complete. The purpose of this review will be to:

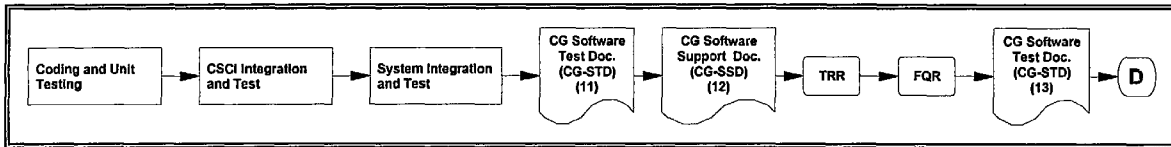
- (1) determine that the detail design of the configuration item under review satisfies the performance and engineering requirements presented in the CG-SRD and CG-SDD,
- (2) determine that the detail design supports compatibility among the configuration item and other items of equipment, facilities, computer software and personnel, and
- (3) assess configuration item risk areas (on a technical, cost, and schedule basis).

For CSCIs, this review will focus on the determination of the acceptability of the detailed design, performance, and test characteristics of the design solution, and on the adequacy of the operation and support documents:

- * Coast Guard Software Plan (CG-SP),
- * Coast Guard Requirements Document (CG-SRD),
- * Coast Guard Software Design Document (CG-SDD),
- * Coast Guard Software Test Plan (CG-STP), and
- * Coast Guard Software User & Operator Manuals (CG-SUOM).

A successful CDR is predicated upon the Coast Guard's determination that the completed *CG Software Design Document (CG-SDD)* and section 4 of the *CG Software Test Document (CG-STD)* form a satisfactory basis for proceeding into *Coding and Testing*.

END DESIGN PHASE



CODING AND TESTING PHASE:

Coding and Unit Testing. Transforming algorithms defined during the detailed design stage into a computer-understandable language. Check each coded module for the presence of bugs. Unit testing's purpose is to ensure that each as-built module behaves according to its specification defined during detailed design.

CSCI Integration and Testing. Interconnecting set of previously tested modules to ensure that the sets behave as well as they did as independently tested modules. Thus, CSCI integration testing's purpose is to ensure that each as-built component behaves according to its specification defined during preliminary design.

System Integration and Test. Check that the entire (i.e., fully integrated) software system embedded in its actual hardware environment behaves according to the *CG Software Requirements Document (CG-SRD)*. **CG Software Test Document (CG-STD)**. A single document encompassing the test planning, test cases and procedures, and test results for CSCI, CSCI integration, and software system testing as applicable. Provides a record of the qualification testing performed on a Computer Software Configuration Item (CSCI), a software subsystem, or other related item. Enables the Coast Guard to assess the testing and its results. The **CG-STD** provides guidance for and comprises the following information: a Software Test Plan, a Software Test Description, and a Software Test Report. (11) *CG-STD sections 1 through 3 completed (with procedures).*

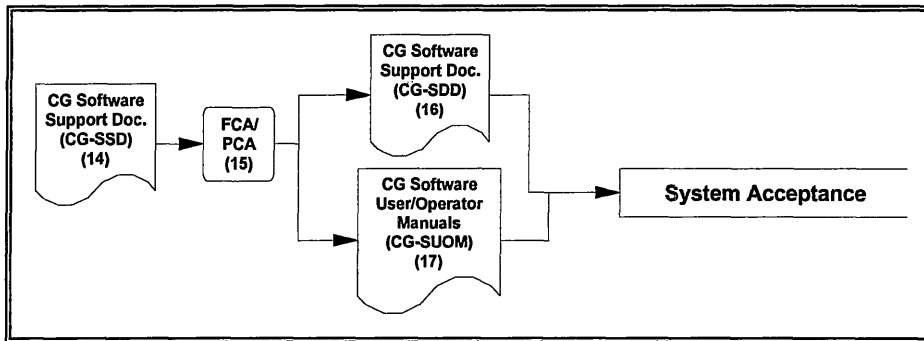
CG Software Support Document (CG-SSD). A single document encompassing the version description, product specification, and support manuals for one or more CSCI(s). (12) *CG-SSD section 4 Software Product Specification.*

Test Readiness Review (TRR). A review conducted for each CSCI to determine whether the software test procedures are complete. Software test procedures are evaluated for compliance with the software test plans and descriptions (CG-STD) and for adequacy in accomplishing test requirements. At TRR, the Coast Guard also reviews the results of software testing and any updates to support documentation.

Formal Qualification Review (FQR). The test, inspection, or analytical process by which a group of configuration items comprising the system are verified to have met specific Coast Guard contractual performance requirements (specifications or equivalent).

CG Software Test Document (CG-STD). A single document encompassing the test planning, test cases and procedures, and test results for CSCI, CSCI integration, and software system testing as applicable. Provides a record of the qualification testing performed on a Computer Software Configuration Item (CSCI), a software subsystem, or other related item. Enables the Coast Guard to assess the testing and its results. The CG-STD provides guidance for and comprises the following information: a Software Test Plan, a Software Test Description, and a Software Test Report. (13) *CG-STD completed with Software Test Report.*

END CODING AND TESTING PHASE



SUPPORT -- OPERATION & MAINTENANCE

All outputs from the previous phases flow into the Support -- Operation & Maintenance Phase. The phase outlines the final steps before system acceptance and ensures that all support documentation has been properly delivered. This phase also supports operations and maintenance by ensuring that all baseline specifications (documentation) will be updated when Change Requests occur.

CG Software Support Document (CG-SSD). A single document encompassing the version description, product specification, and support manuals for one or more CSCI(s). (14) *CG-SSD section 4 (Version Description Documentation) completed.*

Functional Configuration Audit (FCA) / Physical Configuration Audit (PCA). **FCA** - A formal audit to validate that the development of a configuration item has been completed satisfactorily and that the configuration item has achieved the performance and functional characteristics specified in the functional or allocated configuration identification. In addition, the completed operation and support documents shall be reviewed. **PCA** -A technical examination of a designated configuration item to verify that the configuration item "As Built" conforms to the technical documentation which defines the configuration item. (15) *May be held in lieu of, or in addition to, a Software Supportability Review and Software Usability Review.*

CG Software Support Document (CG-SSD). A single document encompassing the version description, product specification, and support manuals for one or more CSCI(s). (16) *CG-SSD completed.*

CG Software User/Operator Manual (CG-SUOM). A single document encompassing all instructions to hands-on users, computer center software operators, preparers and users of software input/output, and operators of computer hardware. The **CG-SUOM** provides guidance for and comprises the following information: a Software User Manual, a Computer Center Software Operator Manual, Software Input/Output Manual, Computer System Operator Manual. (17) *CG Software User/Operator Manual completed.*

System Acceptance.

END SUPPORT -- OPERATION & MAINTENANCE

References :

DoD Harmonization Working Group. Military Standard 498. USA: The Department of Defense, 1994.

DoD Harmonization Working Group. Military Standard 498; Handbook Volume 1. USA: The Department of Defense, 1994.

Davis, Alan M. Software Requirements; Objects, Functions, & States. USA: Prentice-Hall, 1993.

R. H. Thayer and M. Dorfman. System and Software Requirements Engineering. USA: IEEE Computer Society Press, 1990.

SAMPLE GENERIC STATEMENT OF WORK

Referencing the
COAST GUARD SOFTWARE DEVELOPMENT AND DOCUMENTATION
STANDARDS

SECTION C

Table of Contents

| | |
|--|------------|
| C.1 GENERAL | C-1 |
| C.1.1 Scope | C-1 |
| C.1.2 Background | C-1 |
| C.1.3 Contractor Personnel | C-2 |
| C.1.4 Quality | C-2 |
| C.2 DEFINITIONS | C-2 |
| C.3 GOVERNMENT FURNISHED PROPERTY/INFORMATION | C-2 |
| C.4 CONTRACTOR FURNISHED ITEMS | C-2 |
| C.5 SPECIFIC TASKS | C-3 |
| C.5.1 Conversion of the CG Mission Essential Application (CGMEA), CLIN 0001 | C-4 |
| C.5.1.1 Software Development Planning, CLIN 0001A | C-4 |
| C.5.1.2 Systems Requirements Analysis, CLIN 0001B | C-4 |
| C.5.1.3 System Design, CLIN 0001C | C-5 |
| C.5.1.4 Software Requirements Analysis, CLIN 0001D | C-5 |
| C.5.1.5 Preliminary Software Design, CLIN 0001E | C-5 |
| C.5.1.6 Detailed Software Design, CLIN 0001F | C-6 |
| C.5.1.7 Code Development and Unit Testing, CLIN 0001G | C-7 |
| C.5.1.8 CGMEA CSCI Integration Testing, CLIN 0001H | C-7 |
| C.5.1.9 CGMEA System Tuning, CLIN 0001I | C-7 |
| C.5.1.10 System Integration and Testing, Test Readiness Review (TRR), CLIN 0001J | C-7 |
| C.5.1.11 Formal Qualification Testing/ Acceptance, CLIN 0001K | C-7 |
| C.5.2 Data for the CG Mission Essential Application (CGMEA) Conversion, CLIN 0002 | C-8 |
| C.5.2.1 Software Plan (SP), CLIN 0002A | C-8 |
| C.5.2.2 Software Requirements Document (SRD), CLIN 0002B | C-9 |

| | | |
|---------|--|-----|
| C.5.2.3 | Software Design Document (SDD), CLIN 0002C | C-9 |
| C.5.2.4 | Software Test Document (STD), CLIN 0002D | C-9 |
| C.5.2.5 | Software Support Document (SSD), CLIN 0002E | C-9 |
| C.5.2.6 | Computer Software: Converted CG Mission Essential Application (CGMEA), CLIN 0002F | C-9 |

Table of Contents

| | | |
|--------------|---|-------------|
| C.5.3 | CG Mission Essential Application (CGMEA) Legacy data Migration, CLIN 0003 | C-9 |
| C.5.3.1 | Migration Analysis, CLIN 0003A | C-10 |
| C.5.3.2 | Requirements Analysis, CLIN 0003B | C-10 |
| C.5.3.3 | Design/Development of Legacy Data Migration Utility, CLIN 0003C | C-10 |
| C.5.3.4 | Code and Unit Test: Data Migration Utility, CLIN 0003D | C-10 |
| C.5.3.5 | CGMEA Legacy Data Migration Process, CLIN 0003E | C-11 |
| C.5.3.6 | CG Mission Essential Application Legacy Data Migration, CLIN 0003F (optional) | C-11 |
| C.5.4 | Data for CG Mission Essential Application (CGMEA) Legacy Data Migration, CLIN 0004 | C-11 |
| C.5.4.1 | Technical Report on Data Migration Risk Analysis, CLIN 0004A | C-11 |
| C.5.4.2 | Legacy Data Migration Software Requirements Document, CLIN 0004B | C-11 |
| C.5.4.3 | Legacy Data Migration Software Design Document (SDD), CLIN 0004C | C-11 |
| C.5.4.4 | Legacy Data Migration Process Guide, Software User/Operator Manual (SUOM), CLIN 0004D | C-11 |
| C.5.4.5 | CGMEA Legacy Data Migration Utility CSCI, CLIN 0004E | C-12 |
| C.5.5 | Training, CLIN 0005 | C-12 |
| C.5.5.1 | End User Training, CLIN 0005A | C-12 |
| C.5.5.2 | Develop CGMEA Software User and Operators Procedures, CLIN 0005B | C-12 |
| C.5.6 | Data for the CG Mission Essential Application (CGMEA) Training, CLIN 0006 | C-12 |
| C.5.6.1 | CGMEA User/Operator Manual (SUOM), CLIN 0006A | C-12 |
| C.5.7 | Contract Management, CLIN 0007 | C-13 |

| | | |
|---------|--|------|
| C.5.7.1 | Management Process Documentation; Requirements of the Software Plan (SP), CLIN 0007A | C-13 |
| C.5.7.2 | Software Quality Assurance (QA), CLIN 0007B | C-13 |
| C.5.7.3 | Software Configuration Management (SCM), CLIN 0007C | C-13 |
| C.5.7.4 | Technical Interchange Meetings, Reviews and Audits, CLIN 0007D | C-14 |

Table of Contents

| | |
|--|-------------|
| C.5.7 Contract Management, CLIN 0007 (Cont.) | |
| C.5.7.5 Project Status Reporting, CLIN 0007E | C-14 |
| C.5.7.6 Metrics, CLIN 0007F | C-14 |
| C.5.7.7 Government Directed Travel, CLIN 0007G | C-15 |
| C.5.8 Data for Project Management, CLIN 0008 | C-15 |
| C.5.8.1 Monthly Progress Report, CLIN 0008A | C-15 |
| C.5.8.2 Technical Interchange Meeting (TIM) and Review Agenda, CLIN 0008B | C-15 |
| C.5.8.3 Minutes from TIM, Reviews, and Audits. CLIN 0008C | C-15 |
| C.6. APPLICABLE REFERENCE DOCUMENTS | C-16 |
| C.7 CONTRACTOR PERSONNEL REQUIREMENTS | C-16 |
| C.7.1 Key Personnel | C-16 |
| C.7.2 Personnel Qualifications | C-17 |
| C.7.2.1 Contractor Program Manager | C-17 |
| C.7.2.2 Senior Computer System Analyst | C-18 |
| C.7.2.3 Quality Assurance Manager | C-18 |
| C.7.2.4 Computer Systems Analyst | C-19 |
| C.7.2.5 Senior Software Engineer | C-20 |
| C.7.2.6 Application Programmer/Software Engineer | C-21 |
| C.7.2.7 Database Management System Specialist | C-22 |
| C.7.2.8 Documentation Specialist | C-22 |
| C.8. CONTRACTOR SECURITY REQUIREMENTS | C-23 |
| C.8.1 General Security | C-23 |
| C.8.2 Administrative Security - Unclassified Systems | C-24 |
| C.8.3 Malicious Software | C-25 |
| C.9. TRANSITION PLANNING AND SUPPORT SERVICES | C-25 |
| C.10. NOTES | C-25 |
| C.11. PERFORMANCE REQUIREMENTS SUMMARY | C-26 |

PART I, SECTION C

DESCRIPTION/SPECIFICATION/WORKSTATEMENT

C.1 GENERAL

C.1.1 Scope

- 1.1.1 This Statement of Work (SOW) defines the effort required to perform software design, development, testing, documentation, and training for the conversion and integration of the Coast Guard's CG Mission Essential Application (CGMEA), and the migration of legacy CGMEA data from the current USCG Standard Workstation Environment (CGSWII) to that of the next generation Standard Workstation Environment (CGSWIII).
- 1.1.2 The Coast Guard defines Software Conversion as the transformation, without functional change, of computer programs and data elements to permit their use on a replacement or changed hardware/software environment. Thus, this effort includes those tasks necessary to accomplish that conversion.
- 1.1.3 Other objectives within this scope include full documentation of the converted application to adopted standards, an analysis of legacy data migration, the development of data conversion tools and utilities to effect that migration, and training of a select set of management and end user domain experts on the newly delivered system.

C.1.2 Background

- 1.2.1 The U.S. Coast Guard is in the process of procuring a new client/server computing environment under the Coast Guard's Standard Workstation III (CGSWIII) procurement. The CGSWIII procurement complies with Federal Information Processing Standards (FIPS) and includes Intel based workstation and server hardware, and the Windows NT operating system environment. The software environment includes the operating system on all workstation and server platforms, the Informix SE Online Relational Database Management System (RDBMS), the Windows graphical user interface, Powerbuilder

applications development tools and C/C++, Pascal and Ada language tools. As a result, the Coast Guard has identified the CG Mission Essential Application (CGMEA) as one of a number of Mission Essential Applications (MEAs) requiring conversion from the current proprietary operating environment to that of CGSWIII.

- 1.2.2 Documentation and software development processes in support of conversion and integration are to be guided by the Coast Guard Software Development and Documentation Standard (SDDS). This comprehensive standard is based on MIL-STD-498, tailored for the Coast Guard. A full list of referenced documents and provisions as to how to obtain those documents is provided in Section J (Attachment J-1) of this contract.

C.1.3 Contractor Personnel

- a. The Contractor shall provide and maintain personnel qualified to perform the tasks necessary to provide the items and services required by this Contract.
- b. The majority of this effort is to be performed at the Contractor's own facilities.
- c. Additional Contractor Personnel requirements are presented in this Work Statement in Section C.7, Contractor Personnel Requirements.

C.1.4 Quality

- a. The Contractor is responsible for delivering items and services which meet the requirements of this Work Statement.
- b. Additional quality requirements are presented as part of the Contract Management Task.

C.2 DEFINITIONS

Attachment J-2 contains a list of acronyms and definitions for terms used in this Work Statement.

C.3 GOVERNMENT FURNISHED PROPERTY/INFORMATION

- a. Attachment J-3 contains the listing of Government Furnished Property and Information for this Work Statement.
- b. The Contractor shall manage, maintain, and provide accurate accounting for all Government Furnished Property (GFP) and Government Furnished Information (GFI).

C.4 CONTRACTOR FURNISHED ITEMS

Except for property or information specifically stated to be Government furnished in Section C.3, the contractor shall furnish everything required to perform the work described by this Work Statement. Contractor furnished items and services shall be in accordance with the references and standards listed in this Work Statement and to the technical specifications and functional requirements contained in Section J (Attachments).

C.5 SPECIFIC TASKS

The work required by this contract shall be performed in accordance with the listed references, standards, technical specifications and functional requirements contained in Section J (Attachments). A listing of all attachments is provided in Section C.10 Notes, of this work statement. Examples of attachments include the CGMEA Performance Specification (J-4), the source and target environments of the CGMEA application (J-5) and the Development Environment Specification (J-6). The contractor shall ensure that the system delivered is developed to the terms of this contract. Application development and documentation is to be performed in accordance with the Coast Guard Software Development and Documentation Standards (SDDS) and the Coast Guard Software Engineering Guidelines. Extensive reference to processes, activities, data items and deliverables as defined in the SDDS are made in this work statement. These specific references are identified with italicized type. Technical data to be furnished hereunder shall be prepared in accordance with supporting Data Item Descriptions as defined in the SDDS and Section J

(Attachment J-7) to this contract, Contract Data Requirements List (CDRL).

In planning and executing the tasks contained in this DO, the Contractor shall ensure conformance to the following broad requirements:

- a. Ensure transformation of the application into the CGSWIII environment without functional change by providing equivalent functionality, inputs, and outputs.
- b. Maintain essentially the same form and content screens with the addition of GUI functionality.
- c. Ensure converted application produces identical results through stringent testing to include processing comparisons with the existing application.
- d. Emulation of the current CGSWII application environment on CGSWIII is not to be construed as meeting the requirements of this SOW.
- e. Do not construe this conversion and integration effort as requiring a line-for-line conversion.
- f. Ensure preservation and improvement of existing processing efficiencies while improving portability and maintainability of the converted application.
- g. The Contractor shall employ structured analysis and design techniques. Where not otherwise specified, the Yourdon/DeMarco methodology shall be used for modeling the process of information flow and content.

The contractor shall furnish the supplies and services set forth in Section B as follows:

C.5.1 Conversion of the CG Mission Essential Application (CGMEA), CLIN 0001

The Contractor shall develop, document, test, and integrate the CG Mission Essential Application in accordance with the references and standards listed and to the technical specifications and functional requirements contained in Section J (Attachments J-4 through J-6). The Contractor shall ensure that the

correct version of commercial off-the-shelf (COTS) software, as specified in Section J (Attachment J-6), are used for all development, test, and integration activities involving COTS. Specific tasks are as follows:

C.5.1.1 Software Development Planning, CLIN 0001A

The Contractor shall implement a software development process and document the process in a *Software Plan (SP)*. The development of the SP will be guided by the listed references, standards, technical specifications and functional requirements contained in Section J (Attachments). The *SP* shall be reviewed at a *Software Plan Review*.

C.5.1.2 System Requirements Analysis, CLIN 0001B

The Contractor shall perform a *System Requirements Analysis* of the CG Mission Essential Application application specified in Attachment J-4. Activities shall include, but are not limited to, review of the CG Mission Essential Application technical specifications and functional requirements, review of supporting application documentation (furnished as GFI), interviews with Government designated domain experts and system users, and examination of the existing CGMEA application in its current environment (CGSWII). The contractor shall document the results of the requirements analysis in a preliminary *Software Requirements Specification* as defined in Section 5 of the *Software Requirements Document (SRD)*. CGMEA business processes shall be documented in accordance with IDEF0 process modeling practices as specified in FIPS Pub 183. The information shall include a preliminary set of qualification requirements. These sections of the document shall be reviewed at a *Systems Requirements Review*. The Contractor shall update the *SRD* based on the results of the review.

C.5.1.3 System Design, CLIN 0001C

Following completion of the Systems Requirements Review, the Contractor shall develop a System/Segment Design. The proposed design shall be documented in the

Software Design Document (SDD). This document shall be reviewed at a formal *System Design Review*.

C.5.1.4 Software Requirements Analysis, CLIN 0001D

The Contractor shall perform software requirements analysis for the CGMEA Computer Software Configuration Items (CSCI). The Contractor shall document the complete set of software requirements in the *Software Requirements Document (SRD)*. The contractor shall also document the complete set of interface requirements in the SRD. The SRD shall include a complete set of qualification requirements. The *SRD* shall be reviewed at a *Software Requirements Review*. The Contractor shall update the *SRD* based on the results of that review.

C.5.1.5 Preliminary Software Design, CLIN 0001E

The Contractor shall perform preliminary Architecture, Interface and Database software design activities for the CGMEA application. These activities are to be documented in the *Software Design Document (SDD)* in accordance with the approved *Software Plan (SP)*. During the Preliminary Software Design, the Contractor shall initiate development of a *Software Test Document (STD)* and *Software User/Operator Manual (SUOM)*. The Contractor shall establish and maintain Software Development Files (SDFs) which shall be subject to review by the Government. The *SDD*, *STD*, and *SUOM* are subject to formal reviews.

C.5.1.5.1 Architecture. The Contractor shall develop a preliminary software design and document this information in the *SDD*. The Contractor shall design the CGMEA application to conform to the Intermediate Layered Architecture as described in the Coast Guard Software Engineering Guidelines.

C.5.1.5.2 Interface. The Contractor shall develop an interface design in support of the functional requirements specified in the *SRD*. The Contractor shall document the design in the *SDD*.

C.5.1.5.3 Database. The Contractor shall develop a logical database design in support of the

functional requirements specified in the approved *SRD*. The database design shall contain all data entities and relationships necessary to support and maintain the new software and existing data elements. The logical database design shall be compliant with IDEF1X information modeling practices as specified in FIPS Pub 184 and shall be decomposed to third normal form (3NF). Data element naming conventions shall be in compliance with the Coast Guard Data Element Naming Standards as described in COMDTINST M5230.42 (series).

C.5.1.5.4 Software Test Preparation and Software User/Operators Manual (SUOM) Development. During Preliminary Software Design, the Contractor shall develop a Software Test Plan and document this information in the *STD*. In addition, the Contractor shall initiate development of the *SUOM*.

C.5.1.6 Detailed Software Design, CLIN 0001F

Upon review and approval of the preliminary design(s), the Contractor shall proceed with detailed design activities as described below. The Contractor shall update the *Software Design Document (SDD)* to reflect any changes resulting from review of this document by the Government. During detailed design, the Contractor shall develop test cases and shall document them in the *Software Test Document (STD)*.

C.5.1.6.1 Architecture. Upon review and approval of the preliminary design, the Contractor shall develop a detailed system architectural design and document this information in the *SDD*.

C.5.1.6.2 Interface. Upon review and approval of the preliminary interface design, the Contractor shall develop a detailed interface design and document this information in the *SDD*.

C.5.1.6.3 Database. Upon review and approval of the logical database design (3NF), the Contractor shall develop a key-based, fully attributed data model; physical design; and code table specifications and shall document this information in the *SDD*.

C.5.1.7 Code Development and Unit Testing, CLIN 0001G

The Contractor shall perform code development and unit testing on the CGMEA CSCI in accordance with the approved *Software Plan (SP)*, *Software Design Document (SDD)*, and the *Software Test Document (STD)*. The Contractor shall document the results of all testing in CSCI Software Development Folders.

C.5.1.8 CGMEA CSCI Integration Testing, CLIN 0001H

The Contractor shall perform integration and testing on the CGMEA CSCI in accordance with the approved *Software Plan (SP)*, *Software Design Document (SDD)*, and the *Software Test Document (STD)*. The Contractor shall develop test procedures and document plans for testing the system software in the *STD*. The Contractor shall update the *STD* as necessary to reflect changes in test procedures and document test results in the CGMEA CSCI Software Development Folders.

C.5.1.9 CGMEA System Tuning, CLIN 0001I

The Contractor shall tune the converted application to optimize its performance in the target CGSWIII environment. Any modifications made to the application to optimize its performance in the target environment shall be documented in the *Software Support Document (SSD)*.

C.5.1.10 System Integration and Testing, Test Readiness Review (TRR), CLIN 0001J

The Contractor shall draft and deliver Section 4 (Software Product Specification) of the *Software Support Document (SSD)* prior to convening a *Test Readiness Review (TRR)*. The *TRR* shall determine the Contractor's readiness for Formal Qualification Testing (FQT).

C.5.1.11 Formal Qualification Testing/Acceptance, CLIN 0001K

- a. The Contractor shall participate in Formal Qualification Testing (FQT) of each developed CSCI (including the Legacy Data Migration Software developed under CLIN 0003), in accordance with the approved *Software Plan (SP)*, and the *Software Test Document (STD)*. The Contractor shall have received in writing, approvals for the *SP*, *STD* and results of the *Test Readiness Review* prior to the commencement of FQT.
- b. Prior to commencement of FQT, the Coast Guard Contracting Officer will designate in writing Government personnel responsible for conducting FQT. This document will specify a Government Spokesperson as the single point of contact for the Government for matters regarding FQT and will be provided to the contractor in advance of FQT.

C.5.1.11.1 Formal Qualification Testing (FQT).

The Contractor shall conduct FQT utilizing procedures documented in the *Software Plan (SP)*, *Software Test Document (STD)*, and the build processes as documented in the *Software Support Document (SSD)*. The Contractor shall update the *STD* to reflect any changes to tests conducted during FQT. The Contractor shall record the results of the tests and document them in a Software Test Report (Section 5 of the *STD*). The Contractor shall make necessary revisions to application code based on the results of FQT and document those changes in the *Software Design Document (SDD)*. The Contractor shall document the "as built" baseline (Software Product Specification) and identify the exact version of the delivered software (Version Description) in the *SSD*. The Government reserves the right to convene a *Software Supportability Review* to determine the adequacy of the documentation to support the software.

C.5.1.11.2 Acceptance Criteria. The purpose of FQT is to verify and validate conformance of the converted CG Mission Essential Application software to the requirements specified in the approved *Software Requirements Document (SRD)*. Acceptance Criteria are specified in Section E of

this contract.

C.5.2 Data for the CG Mission Essential Application CGMEA Conversion, CLIN 0002

Technical data to be furnished hereunder shall be prepared in accordance with supporting Data Item Descriptions as defined in the CG-SDDS and Section J (Attachment J-7), Contract Data Requirements List (CDRL).

C.5.2.1 Software Plan (SP), CLIN 0002A, [CDRL A001]

This Data Item describes format, contents and preparation instructions for data generated in support of software development planning.

C.5.2.2 Software Requirements Document (SRD), CLIN 0002B, [CDRL A002]

This Data Item describes format, contents and preparation instructions for data generated in support of system and software requirements analysis.

C.5.2.3 Software Design Document (SDD), CLIN 0002C, [CDRL A003]

This Data Item describes format, contents and preparation instructions for data generated in support of system and software design.

C.5.2.4 Software Test Document (STD), CLIN 0002D, [CDRL A004]

This Data Item describes format, contents and preparation instructions for data generated in support of code, unit, integration and system testing.

C.5.2.5 Software Support Document (SSD), CLIN 0002E, [CDRL A005]

This Data Item describes format, contents and preparation instructions for data generated in support of delivered software CSCIs.

C.5.2.6 Computer Software: Converted CG Mission Essential Application CGMEA, CLIN 0002F

[CDRL A006]

The delivered CG Mission Essential Application CSCI shall include source and executable code and any support files necessary to compile, link, or execute the software. The CSCI shall be delivered on magnetic media compatible with CGSWIII hardware and software as specified in Section J (Attachment J-5, Application Environment Specification). Any subsequent delivery shall include a summary report which details any software changes from the previous delivery.

C.5.3 CG Mission Essential Application(CGMEA) Legacy Data Migration, CLIN 0003

Under Government direction, perform an analysis on CGMEA legacy data. Develop, test, and document a data migration process and required automated data migration utilities, and perform CGMEA legacy data migration as specified below.

C.5.3.1 Data Migration Analysis, CLIN 0003A

The Contractor shall perform an analysis of the CGMEA legacy data conversion requirements. The Contractor shall examine the magnitude, integrity and complexity of the existing legacy database(s) to ascertain impact and risks associated with a fully automated conversion, reentry of data, or no conversion. The Contractor shall consider such requirements as mapping routines and integrity checks and the possibility of different default data types in the analysis. The analysis shall be summarized in a Pro/Con analysis followed by recommendations and justification along with any and all cost estimates. The Contractor shall document the results of the analysis in a Technical Report.

C.5.3.2 Requirements Analysis, CLIN 0003B

The Contractor shall perform System Requirements Analysis activities for the CGMEA Data Migration Utility CSCI. The Contractor shall document a complete set of software requirements in the *Software Requirements Document (SRD)*. The *SRD* shall be reviewed at the Software Requirements Review (SWRR). The

Contractor shall update the SRD based on the results of the review.

C.5.3.3 Design/Development of Legacy Data Migration Utility, CLIN 0003C

The Contractor shall design and develop an automated data migration utility for use at multiple conversion sites. The Contractor shall analyze existing data structures, define the transformation algorithms to be used to convert the existing data elements to the new structure, define the resulting data elements, and design the software required to perform the transformations and populate the target database(s) and files. The Contractor shall document the design in the *Software Design Document (SDD)*.

C.5.3.4 Code and Unit Test: Data Migration Utility, CLIN 0003D

The Contractor shall perform code and unit testing on the developed data migration utility CSCI in accordance with the approved *Software Plan (SP)*, *Software Design Document (SDD)*, and the *Software Test Document (STD)*. This CLIN is subject to FQT IAW Para 5.1.10. The Contractor shall document the results of all testing in the CSCI Software Development Folders.

C.5.3.5 CGMEA Legacy Data Migration Process, CLIN 0003E

The Contractor shall develop a detailed process for migration of application legacy database(s) and files from the application's current environment to the new application environment. The Contractor shall develop detailed instructions for use by Coast Guard personnel detailing step-by-step procedures for accomplishing data migration from the CGMEA source environment to that of the converted CGMEA applications environment. The Contractor shall document this process in the *Software/User Operator Manual (SUOM)*. The migration process shall be reviewed at the *Software Requirements Review (SWRR)*. The Contractor shall update the *SUOM* based on the results of the review.

C.5.3.6 CG Mission Essential Application Legacy Data Migration, CLIN 003F (OPTIONAL)

The Contractor shall perform CGMEA legacy data migration, as directed by the Government, at designated Government sites as specified in Section J (Attachment J-8). This data migration shall be performed in accordance with the approved migration process developed and documented in accordance with CLIN 0003E.

C.5.4 Data for CG Mission Essential Application (CGMEA) Legacy Data Migration, CLIN 0004

Technical data to be furnished hereunder shall be prepared in accordance with Data Item Descriptions (DIDs) as specified in the CG-SDDS and Section J (Attachment J-7), Contract Data Requirements List (CDRL).

C.5.4.1 Technical Report on Data Migration Analysis, CLIN 0004A [CDRL A007]

C.5.4.2 Legacy Data Migration Software Requirements Document (SRD) (Section 5, Incorporate into CLIN 0002B), CLIN 0004B [CDRL A002]

C.5.4.3 Legacy Data Migration Software Design Document (SDD) (Section 4, Incorporate into CLIN 0002C), CLIN 0004C [CDRL A003]

C.5.4.4 Legacy Data Migration Process Guide, Software User/Operator Manual (SUOM) (Incorporate into CLIN 0006A), CLIN 0004D [CDRL A012]

C.5.4.5 CGMEA Legacy Data Migration Utility CSCI, CLIN 0004E [CDRL A011]

The delivered CG Mission Essential Application Legacy Data Migration Utility CSCI shall include source and executable code and any support files necessary to compile, link, or execute the software. The CSCI shall be delivered on magnetic media compatible with CGSWII or CGSWIII hardware and software as specified in Section J (Attachment J-5), Application Environment Specification. Any subsequent delivery shall include a

summary report which details any software changes from the previous delivery.

C.5.5 Training, CLIN 0005

The Contractor shall provide training and develop training material as specified below and in Section J (Attachment J-9, Section 3.2) of this Statement of Work. The Contractor developed training material shall be incorporated as an appendix to the *Software User and Operator Manual*.

C.5.5.1 End User Training, CLIN 0005A

The Contractor shall provide hands-on training to designated Government personnel as specified in Section J (Attachment J-9).

C.5.5.2 Develop CGMEA Software User and Operators Procedures, CLIN 0005B

The Contractor shall develop procedures, operating guides, and job-aids for end-users, computer center software operators, preparers and users of application inputs/outputs, and operators of computer hardware. These procedures shall be documented in the *Software User/Operators Manual (SUOM)* and shall be reviewed for acceptance at a *Software Usability Review*.

C.5.6 Data for the CG Mission Essential Application (CGMEA) Training, CLIN 0006

Technical data to be furnished hereunder shall be prepared in accordance with Data Item Descriptions (DIDs) as specified in the CG-SDDS and Section J (Attachment J-7), Contract Data Requirements List (CDRL).

C.5.6.1 CGMEA User/Operator Manual (SUOM), CLIN 0006A, [CDRL A012]

This Data Item describes format, contents and preparation instructions in support of providing CGMEA applications specific instruction to hands-on users and computer center software operators.

C.5.7 Contract Management, CLIN 0007

- a. The Contractor shall establish and maintain management operations and procedures necessary to control the total software engineering effort and assure effective cost and schedule control of the conversion effort. The Contractor shall document their process and perform quality assurance, configuration management, operational support activities, and participate in reviews and audits as necessary to accomplish the work defined in this SOW.
- b. Contractor shall designate a Program Manager (PM) to perform the duties as specified in Section C.7 Contractor Personnel Requirements.

C.5.7.1 Management Process Documentation; Requirements of the Software Plan (SP), CLIN 0007A [CDRL A001]

The Contractor shall establish and implement a program management office function to manage all technical performance, cost, schedule, and data delivery requirements of the contract. The contractor shall document this process in the *Software Plan (SP)*.

C.5.7.2 Software Quality Assurance (QA), CLIN 0007B

The contractor shall implement procedures to comply with the software quality requirements of the *SDDS*. The Contractor's Quality Assurance approach is to be documented in the *Software Plan (SP)* as approved and accepted by the Government. This documentation shall constitute a Software Quality Assurance Plan for the purposes of this contract. The contractor may include any formally documented in-house Quality Assurance Plans and procedures.

C.5.7.3 Software Configuration Management (SCM), CLIN 0007C

The contractor shall implement procedures to comply with the software configuration management requirements of the *SDDS*. The Contractor's SCM approach is to be

documented in the *Software Plan (SP)* as approved and accepted by the Government. This documentation shall constitute a Configuration Management Plan for the purposes of this contract. The contractor may include any formally documented in-house Software Configuration Management Plans and procedures.

C.5.7.4 Technical Interchange Meetings, Reviews and Audits, CLIN 0007D

The Contractor shall plan for and participate in Joint Technical and Management Reviews and Audits as defined in the SDDS to the level specified in Section J (Attachment J-10) of this contract. The contractor shall schedule formal technical interchange meetings (TIMs) as appropriate throughout this effort.

C.5.7.4.1 Review/Audit Scheduling.

The Contractor shall include in the project schedule all reviews and audits specified in Section J (Attachment J-10) to this contract. Reviews and audits shall coincide with completion of the appropriate documentation and development phases as detailed in the *SDDS* and this *SOW*.

C.5.7.4.2 Documentation of Reviews and Audits.

The contractor shall be responsible for submitting agendas in advance of and documenting and submitting minutes resulting from all scheduled TIMs, reviews and audits.

C.5.7.4.3 Independent Verification and Validation (IV&V).

The Government reserves the right to exercise IV&V via Government team or separate contractor to assure contract compliance. In so doing, the Government stipulates that IV&V activities will be conducted as a function of scheduled reviews and audits contained in this contract. If directed, the Contractor shall assist as necessary the IV&V team/contractor during scheduled reviews and audits.

C.5.7.5 Project Status Reporting, CLIN 0007E

The Contractor shall prepare and deliver monthly project status reports in accordance with *Exhibit GG* of the Software Development and Documentation Standards (SDDS).

C.5.7.6 Metrics, CLIN 0007F

The contractor shall implement procedures to comply with the management indicator (metric) reporting requirements of the SDDS. The Contractor's management approach to collection and application of metrics is to be documented in the *Software Plan (SP)* as approved and accepted by the Government. At a minimum, the Contractor shall prepare and include in the monthly project status reports, Software Management Indicators (Metrics) listed in Appendix F of the SDDS. In addition, the Contractor may include any formally documented in-house metrics deemed appropriate.

C.5.7.7 Government Directed Travel, CLIN 0007G

The Contractor shall *****G-ACS recommended wording for Government directed travel goes here***** It is anticipated that contractor personnel may require travel in conjunction with this effort to government facilities outside the Washington Metropolitan area. However, the Contractor shall make every effort to minimize such travel and the number of personnel involved. Facilities anticipated for visit include _____ located in _____, and the _____ located in _____. Travel is to be conducted in accordance with Section H.x.

C.5.8 Data for Project Management, CLIN 0008

Technical data to be furnished hereunder shall be prepared in accordance with Data Item Descriptions (DIDs) as specified in the CG-SDDS and Section J (Attachment J-7), Contract Data Requirements List (CDRL).

C.5.8.1 Monthly Progress Report, CLIN 0008A [CDRL A013]

This Data Item describes format, contents and preparation instructions for progress reports required in support of project management.

C.5.8.2 Technical Interchange Meeting (TIM) and Review Agenda, CLIN 0008B [CDRL A014]

This Data Item describes format, contents and preparation instructions for data generated in support of Technical Interchange Meetings and Reviews.

C.5.8.3 Minutes from TIM, Reviews, and Audits, CLIN 0008C [CDRL A015]

This Data Item describes format, contents and preparation instructions for data generated in support of Technical Interchange Meetings and Reviews.

C.6 APPLICABLE REFERENCE DOCUMENTS

A full list of referenced documents and provisions as to how to obtain those documents are provided in Section J (Attachment J-1) to this contract.

C.7 CONTRACTOR PERSONNEL REQUIREMENTS

- a. The Contractor shall provide and maintain personnel qualified to perform the tasks necessary to deliver the items and services required by this Work Statement.
- b. The Contractor shall supply personnel who are citizens of the United States, with the only exception being legal residents of the United States who may perform duties with ADP-I (non-critical, non-sensitive) or ADP-O (non-ADP) requirements. COMDTINST M5520.12, Coast Guard Personnel Security Program, and DOT Order 1630.5 define the specific criteria for assigning positions (Government and Contractor) to ADP-IV through ADP-I sensitive categories.
- c. The Contractor shall ensure that a stable, competent work force is selected to perform the tasks associated with this Work Statement. All Contractor personnel are required to read, write, communicate and understand

English fluently unless waived in writing on an individual basis by the Contracting Officer. This Work Statement requires performance by persons, who, over the Contract life, increasingly know the USCG organization, missions, IRM architecture, and policies, and its software and hardware environments.

C.7.1 Key Personnel

- a. Key personnel for this Contract are the Contractor Program Manager, Senior Computer Systems Analyst, and Quality Assurance Manager.
- b. Key personnel shall be substituted in accordance with **Section H** of this contract.
- c. Contractor personnel designated as key personnel per **Section H** of this contract will primarily work at the Contractor's facility where the actual development is being performed unless the Contracting Officer grants an individual-based waiver of this requirement.

C.7.2 Personnel Qualifications

Qualifications for Contractor personnel are further detailed below. While separately specified, in all instances, general, specialized, and specific experience may have been gained concurrently.

C.7.2.1 Contractor Program Manager (Key Personnel)

- a. Duties. Serves as the Contractor's program manager, and shall be the Contractor's authorized interface with the Government Contracting Officer (KO), the Contracting Officer's Technical Representative (COTR), and Government management personnel. Responsible for formulating and enforcing work standards, assigning contractor schedules, reviewing work discrepancies, supervising contractor personnel and communicating policies, purposes, and goals of the organization to subordinates. Shall be responsible for the overall contract performance and shall not serve in any other capacity under this contract.

- b. Qualifications. A Bachelor's degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline. This position requires a minimum of ten years experience, of which at least six years must be specialized.

Specialized experience to include complete engineering project development from inception to deployment, demonstrated ability to provide guidance and direction in tasks similar to the tasks provided in this Statement of Work, proven expertise in the management and control of funds and resources, and demonstrated capability in managing multi-task contracts of this type and complexity. At least one of these years shall have been with the proposed software development environment as specified in Section J (Attachment J-6). General experience includes increasing responsibilities in information systems design and management.

A Master of Science degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline will be considered equivalent to one year specialized experience and two years general experience. A PhD (in the fields described in this paragraph) will be considered equivalent to two years specialized experience and three years general experience.

The Program Manager shall possess a SECRET security clearance. (application specific)

C.7.2.2 Senior Computer System Analyst (Key Personnel)

- a. Duties. Provides technical and administrative direction for personnel performing software development tasks, including the review of work products for correctness, adherence to the design concept and to user standards, and for progress in accordance with schedules. Coordinates with the Program Manager to ensure problem solution and user satisfaction. Makes recommendations, if needed, for approval of major system installations. Prepares milestone status reports, deliverables, and presentations on the system concept to colleagues, subordinates, and end user representatives. Provides daily supervision and direction to support staff.

- b. Qualifications. A Bachelor's degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline. This position requires a minimum of eight years experience, of which at least five years must be specialized. Specialized experience to include analysis and design of complex business applications utilizing database management systems and programming languages specified in Section J (Attachment J-6). At least one of these years shall have been with the proposed software development environment as specified in Section J (Attachment J-6). Knowledge of current storage and retrieval methods, graphical client/server development, and demonstrated ability to formulate specifications for computer programmers to use in coding, testing and debugging of computer programs. General experience includes increasing responsibilities in assignments of a technical nature. Proven ability to work independently or under only general direction on complex application problems involving all phases of system analysis is required.

A Master of Science degree in Computer Science or Information Systems will be considered equivalent to one year specialized experience and two years general experience. A PhD in Computer Science or Information Systems will be considered equivalent to two years specialized experience and three years general experience.

C.7.2.3 Quality Assurance Manager (Key Personnel)

- a. Duties. Establishes and maintains a process for evaluating software products, development processes, and associated documentation. Determines the resources for quality control. Maintains a high degree of quality throughout the software development life-cycle (SDLC). Conducts formal and informal reviews at pre-determined points throughout the SDLC. Responsible for configuration management processes, software metrics, and software testing and integration.
- b. Qualifications. A Bachelor's degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline. This

position requires a minimum of six years experience, of which at least three years must be specialized.

Specialized experience to include configuration management, verification and validation, software testing and integration, software metrics and their application to software quality assessment. General experience includes increasing responsibilities in quality assurance and quality control.

A Master of Science degree in Computer Science or Information Systems will be considered equivalent to one year specialized experience and two years general experience.

C.7.2.4 Computer Systems Analyst

- a. Duties. Analyzes and develops computer software possessing a wide range of capabilities, including numerous engineering, business, and records management functions. Develops plans for automated information systems from project inception to conclusion. Analyzes the business problem, and develops system requirements and program specifications, from which programmers prepare detailed information models, function and process models, data flows, programs, and tests. Coordinates closely with programmers to ensure proper implementation of program and system specifications. Develops, in conjunction with functional users, system alternative solutions.
- b. Qualifications. A Bachelor's degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline. This position requires a minimum of six years experience, of which at least four years must be specialized.

Specialized experience to include analysis and design of complex business applications utilizing database management systems and programming languages specified in Section J (Attachment J-6). At least one of these years shall have been with the proposed software development and RDBMS environment as specified in Section J (Attachment J-6). Knowledge of current storage and retrieval methods, one year of system analysis experience designing graphical business applications in a client/server environment.

Demonstrated ability to formulate specifications using IDEF1X and IDEF0 modeling techniques for computer programmers to use in coding, testing and debugging and maintaining computer programs. General experience includes increasing responsibilities in information systems design and management. Must demonstrate the ability to work independently or under only general direction on requirements that are moderately complex to analyze, plan, program and implement.

A Master of Science degree (in the fields described in this paragraph) will be considered equivalent to one year specialized experience and two years general experience. A PhD in Computer Science or Information Systems will be considered equivalent to two years specialized experience and three years general experience.

C.7.2.5 Senior Software Engineer

- a. Duties. Analyzes complex system and software requirements. Designs and/or implements software tools and subsystems to support software reuse and domain analyses. Manages software development and support using formal specifications, information and process models, other accepted design techniques and Computer Aided Software Engineering (CASE) tools. Estimates software development costs and schedules. Reviews existing and developed programs and assists in making refinements, reducing operating time, and improving current techniques. Supervises software configuration management.
- b. Qualifications. A Bachelor's degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline. This position requires a minimum of six years experience, of which at least three years must be specialized.

Specialized experience includes: demonstrated experience in development of large-scale database management systems in a graphical client/server computing environment, working with SQL and third and fourth generation programming languages. At least one of these years shall have been with the proposed software development and RDBMS environment as specified

in Section J (Attachment J-6). Must demonstrate the ability to develop complex software to satisfy design objectives, ability to work independently or under only general direction.

A Master of Science degree in Computer Science, Information Systems or other related scientific or technical discipline will be considered equivalent to one year specialized experience and two years general experience.

C.7.2.6 Application Programmer/Software Engineer

- a. Duties. Analyzes functional business applications and design specifications for functional areas such as accounting, finance, and logistics. Develops block diagrams, information and process models, and logic flow charts. Translates detailed design into computer software. Tests, debugs, and refines the computer software to produce the required product. Prepares required documentation, including both program-level and user-level documentation. Enhances software to reduce operating time and resource requirements, and improve efficiency. Provides technical direction to programmers as required to ensure program deadlines are met.

- b. Qualifications. A Bachelor's degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline. This position requires a minimum of five years experience, of which at least three years must be specialized. Specialized experience to include experience as an applications programmer on large-scale database management systems in a graphical client/server computing environment, knowledge and experience in developing applications to the target CGSWIII application environment as specified in Section J (Attachment J-5). At least one of these years shall have been with the proposed software development and RDBMS environment as specified in Section J (Attachment J-6). Must demonstrate the ability to develop complex software to satisfy design objectives, ability to work independently or under only general direction.

A Master of Science degree in Computer Science, Information Systems or other related scientific or technical discipline will be considered equivalent to one year specialized experience and two years general experience.

C.7.2.7 Database Management System Specialist

- a. Duties. Manages development of complex database projects. Provides highly technical expertise in the use of relational database management systems. Develops and documents logical and physical database structures to support application development to meet validated user requirements. Defines file organizations, indexing methods, security procedures, and database designs that optimize application performance for specific user applications.

- b. Qualifications. A Bachelor's degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline. This position requires a minimum of six years experience, of which at least three years must be specialized. Specialized experience to include demonstrated experience using current RDBMS technologies, application design utilizing various RDBMS and experience with RDBSM internals in graphical, client/server environments. At least one of these years shall have been with the proposed software development and RDBMS environment as specified in Section J (Attachment J-6). General experience includes increasing responsibilities in RDBMS systems analysis and programming. Must demonstrate the ability to develop complex software to satisfy design objectives, ability to work independently or under only general direction.

A Master of Science degree in Computer Science, Information Systems or other related scientific or technical discipline will be considered equivalent to one year specialized experience and two years general experience.

C.7.2.8 Documentation Specialist

- a. Duties. Gathers, analyzes, and composes technical information. Conducts research and ensures the use of proper technical terminology. Translates technical information into clear, readable documents to be used by technical and non-technical personnel. Prepares functional specifications, system specifications, user manuals, special reports and other customer deliverables and documents.
- b. Qualifications. A Bachelor's degree in English, Literature, or other related discipline applicable to this position. This position requires a minimum of three years experience, of which at least one year must be specialized. Specialized experience to include demonstrated experience in preparing technical documentation, technical writing and documentation experience pertaining to all aspects of information systems. Must demonstrate ability to work independently or under only general direction.

C.8 CONTRACTOR SECURITY REQUIREMENTS

C.8.1 General Security

- a. Contract Security Classification Specification (DD Form 254) is applicable and is attached as Attachment J-11.
- b. The Contractor shall comply with the requirements of OMB Circular A-130 in exercising security measures for automated data processing, computer systems and access to the data and facilities.
- c. The Contractor shall provide new systems security and telecommunications network security by following the guidance provided in Chapters 5, 8, 9, 10, and 19 of COMDTINST M5500.13A, and the Privacy Act guidance of COMDTINST M5260.2, Privacy and Freedom of Information Act Manual.
- d. The information systems being built to satisfy the SOW will be unclassified, sensitive information systems. All information or data that they will process, transmit and store will be UNCLASSIFIED. There will be no national security classified data

processed, transmitted or stored. Most of the information will be sensitive data for many reasons, to include FOR OFFICIAL USE ONLY USCG operational, commercial proprietary, investigation sensitive, and Privacy Act. This sensitive data, however, will be used by a few business functions that perform national security-related classified activities. The need to understand the classified aspects of these business functions, in order to properly design these unclassified, sensitive information systems, is what causes the national security personnel and facility requirements of this Contract. Some Task Orders may require access to classified information. Classified information access and security requirements will be furnished on a DD-254, Contract Security Classification Specification, incorporated into those Task Orders requiring access to classified information. Facility and staff security clearances, necessary for performance of a Task Order with a DD-254, shall be in place prior to issuance of a Task Order under this Contract.

- e. The Contractor shall comply with DOD 5220.22-M, Industrial Security Manual for Safeguarding Classified Information, for all matters pertaining to classified information and facility security requirements.
- f. The Contractor shall comply with the Coast Guard Physical Security Manual, COMDTINST M5530.1, for physical security requirements whenever DOD 5220.22-M does not apply.
- g. The Contractor shall comply with the security regulations of the Government installation or Contractor (or subcontractor) facility where on-site work is to be performed.
- h. The Contractor shall not use any electronic information processing equipment in its possession for the purpose of processing or transmitting classified information under this Contract without the written permission of the Contracting Officer.
- i. The Contractor shall not allow employees to use files for logging onto systems that contain the

employee's passwords or access controls. The Contractor shall report discovered instances to the COTR and system administrator immediately.

- j. The Contractor shall obtain written approval from the Contracting Officer prior to introducing Contractor owned information system, photo reproduction, telecommunication and data equipment into a Government site. Contractor owned equipment located within a Government site shall be subject to Government security requirements at that site.

C.8.2 Administrative Security - Unclassified Systems

- a. The Contractor shall follow the control principles described in Chapter 7 of COMDTINST M5500.13A, AIS Security Manual. Administrative security consists of management constraints, standard operational security procedures, security accountability controls, and those additional administrative actions necessary to protect IS resources. Administrative security controls define who, what, where, how, and how often actions are taken in the work flow process to ensure the integrity and confidentiality of IS resources. Emphasis shall be placed at those points in the process where control passes from one function, element, or individual to another, as that is where control can be lost.

C.8.3 Malicious Software

- a. The Contractor is specifically prohibited from designing elements into any software which will disable the software in some way, unless required in the functional description. Malicious code set to the conclusion of the warranty or maintenance option shall be precluded for both custom-developed and off-the-shelf portions of the developed software. Delivery by the Contractor to the Government of any software under this Contract and its Task Orders is under an implied certification by the Contractor that the software does not contain malicious code.

C.9 TRANSITION PLANNING AND SUPPORT SERVICES

- a. Software Warranty. All Contractor delivered software

associated with this Statement of Work shall be warranted for a period of one year from the date of Government acceptance. All software shall satisfy requirements as specified in the solicitation and the Contractor's accepted Software Plan. Any software deficiencies discovered during the warranty period will be reported to the Contractor in writing. The Contractor shall respond to the deficiencies within 15 working days of notification by the Government. The Contractor shall assess the deficiencies and submit in writing to the Government a proposal for corrective actions. The proposal will be approved by the Government before implementation and all software warranty work by the Contractor shall be at no cost to the Government.

(***placeholder***may be better in Sec H)**

- b. Upon completion of development and successful acceptance testing, the developed CG Mission Essential Application and supporting CSCIs and documentation will transition into the Government's configuration management, operation and maintenance phase. The Contractor shall cooperate fully in any transition of developed software components and associated deliverables to follow-on operations and maintenance activities as directed by the Government.

C.10 NOTES

C.10.1 The following is a list of attachments contained in Section J to this contract.

- a. Attachment J-1 contains a full list of referenced documents and provisions as to how to obtain those documents.
- b. Attachment J-2 contains the definitions of important terms and a listing of acronyms used in this Contract.
- c. Attachment J-3 specifies Government Furnished Property and Information.
- d. Attachment J-4 contains the Application Performance Specification (including functional requirements) of the CG Mission Essential Application (CGMEA) Software to be converted under this Work Statement.

- e. Attachment J-5 contains the Applications Environment Specification which describes the current source and target environment of the CG Mission Essential Application.
- f. Attachment J-6 contains the Development Environment Specification which describes software engineering guidelines and constraints, authorized COTS development languages, target RDBMS for development of the CG Mission Essential Application.
- g. Attachment J-7 lists the Contract Data Requirements List (CDRL) for each data item required of this Statement of Work.
- h. Attachment J-8 specifies Coast Guard sites requiring Government directed, Contractor provided legacy data migration services. **(optional)**
- i. Attachment J-9 contains the specific tasking for Section C.5.5, CLIN 0005 Training.
- j. Attachment J-10 lists the mandatory reviews and audits associated with this Work Statement.
- k. Attachment J-11 contains the Contract Security Classification Specification (DD Form 254).

C.11 PERFORMANCE REQUIREMENTS SUMMARY

- a. Major tasks include the actual conversion of the existing application, legacy data migration, training, and contract management. In addition to the actual accomplishment of the tasks themselves, each task has associated documentation or data deliverables.
- b. The contractor will be testing and integrating into Government provided Operational Test & Evaluation (OT&E) and Run-Time Environments. Details on these environments are provided in Attachment J-5.

US Coast Guard
Software Development and Documentation Standards
(CG-SDDS COMDTINST 5234.4)
Guide to Using the Consolidated Software
Development Document
April 1996
US Coast Guard Headquarters
Software Policy & Standards Division (G-SCC-2)
2100 Second Street SW
Washington, DC 20593-0001

Overview

Purpose: The purpose of this guide to using the Coast Guard Software Development and Documentation Standards (CG-SDDS) is to provide an overview of applying the standard to a software development effort. The CG-SDDS is a tasking document that is supported by eight Data Item Descriptions (DIDs). This guide focuses on using the CG Consolidated Software Development Document (CG-CSDD) which is a single DID that describes all aspects of a software development project. The CG-SDDS supports a process-driven software development. A frequent complaint about software development under military standards is that it is document driven. Rather than focusing on the real work to be done, the developer must generate an endless series of documents, some of which have little to do with getting the job done. The contractor may have to translate real work products, such as data in computer-aided software engineering (CASE) tools, into paper documents or translate work products from the actual format in which work is being done to an artificial format imposed by Data Item Descriptions (DIDs).

CG-CSDD DID as an activity checklist. A unique feature of CG-CSDD, when using the consolidated approach, is that it uses sections of the CG-CSDD DID to fully define many of the activities in the standard. The CG-CSDD DID specifies the information that is to be defined and recorded in carrying out the activities. This use of the CG-CSDD DID applies regardless of whether the activities result in a tangible deliverable. CG-SDDS requires work to take place, regardless of whether the resuand gives the Coast Guard access to the results in the developer's facility. When deliverables are needed, they are called in for the CDRL.

Many of the activities in CG-SDDS rely on DIDs to fully define them. For example, the activity of preparing the CG Software Plan (CG-SP) consists of defining and recording all information called for by the CG-SP DID. This requirement applies regardless of whether the CG-SP is in the form of a document and regardless of whether the resulting plan is deliverable. The DID is used as a checklist of information to be defined in carrying out the required activity. This is a new role for DIDs.

Figure 1 illustrates the role of DIDs and software products in CG-SDDS, emphasizing the distinction made between natural work products, the form in which they reside, and their status as deliverables or non-deliverables.

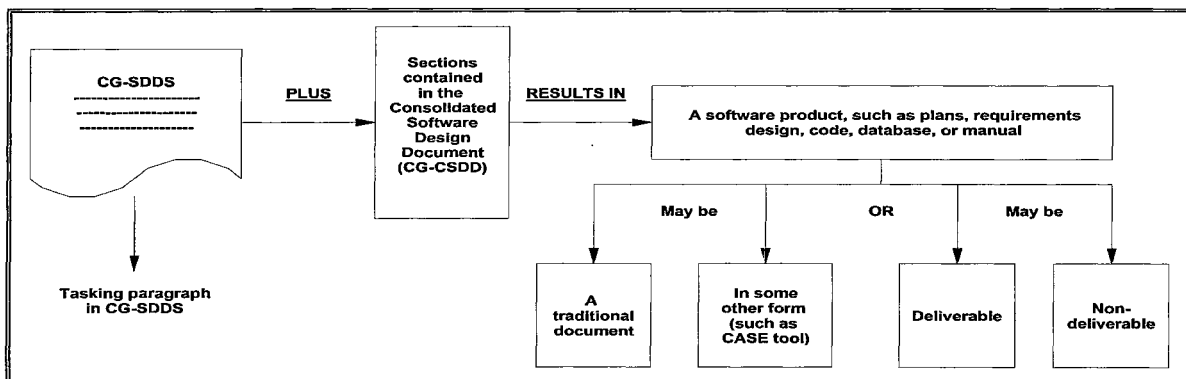


Figure 1: The role of DIDs and software products in CG-SDDS.

Roles of plans in CG-SDDS. The basic philosophy of CG-SDDS is that the standard tells "what" the contractor must do. The developer proposes the "how" via the plans. The Coast Guard has the right to review and approve (or disapprove) the plans, and the

contractor must adhere to the plans once approved. This philosophy makes the plans, especially the Software Plan (Sections 3.1, 3.2, and 3.3), key documents in the project.

CG-SDDS philosophy regarding deliverables. CG-SDDS has been worded to differentiate between the planning and engineering activities that make up a software development project and the generation of deliverables. A key objective of this wording is to eliminate the notion that the Coast Guard must order a given deliverable in order to have planning or engineering work take place. Under the CG-SDDS, the planning and engineering work takes place regardless of which deliverables are ordered, unless a given activity is tailored out of the standard. In addition, joint technical reviews have been included to review the results of that work in its natural form, without the generation of deliverables.

Software products should be designated as deliverables only when there is a genuine need to have planning or engineering information transformed into a deliverable, recognizing that this transformation requires time and effort that would otherwise be spent on the engineering effort.

Multiple formats for deliverables. Traditional deliverables take the form of paper documents exactly following DID formats. While this form works well for some deliverables, it is not the only form, and alternatives should be considered. One variation from paper documents is electronic word processing files containing these documents. This format saves paper, but still requires the contractor to format the information as required by the DID. Another variation is specifying that a paper or word processor document is to include all DID contents but may be in the developer's format. Yet another variation is allowing deliverables to take forms that are not traditional documents at all, such as data in computer-aided software engineering (CASE) tools. These variations can minimize the time spent transforming actual work products into deliverables.

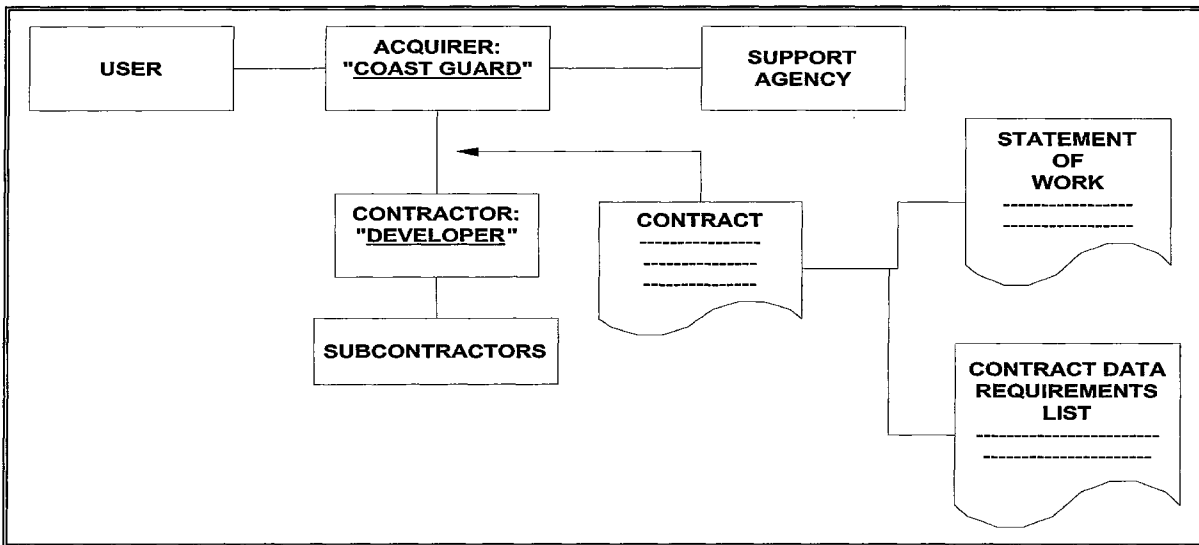


FIGURE 2: Players and agreements assumed by the CG-SDDS

CG-CSDD DID Sections in Relation to the CG-SDDS DIDs. Figure 3 provides a graphical depiction of the CG-SDDS and section descriptions for the CG-CSDD DID. Figure 4 provides a graphical depiction of the functional activities associated with applying the CG-SDDS and the CG-CSDD DID to a software development lifecycle. Following Figure 4 is a one page step-by-step listing of key product reviews and completions of specific sections and

documents (deliverables) generated by the CG-SDDS. Figure 5 is a crosswalk of the CG-CSDD sections to the related CG-SDDS DIDs. Since in many cases the specific CG-SDDS DIDs give clarifying information to what is in the related sections of the CG-CSDD DID, this crosswalk is provided to ease identifying the appropriate related CG-SDDS DID. **However, inclusion of this crosswalk does not obligate the developer to the use of the more detailed DIDs.** Following Figure 5 are several pages with detailed descriptions of each process, document, and review shown in Figure 4. The detailed descriptions are listed in the order and life cycle phase in which they occur.

Applying the CG-CSDD DID: Appendix A provides a sample statement of work (SOW) which contains language for applying the CG-SDDS and CG-CSDD DID to a software development project.

FIGURE 3: Consolidated Software Development Document - Section Descriptions

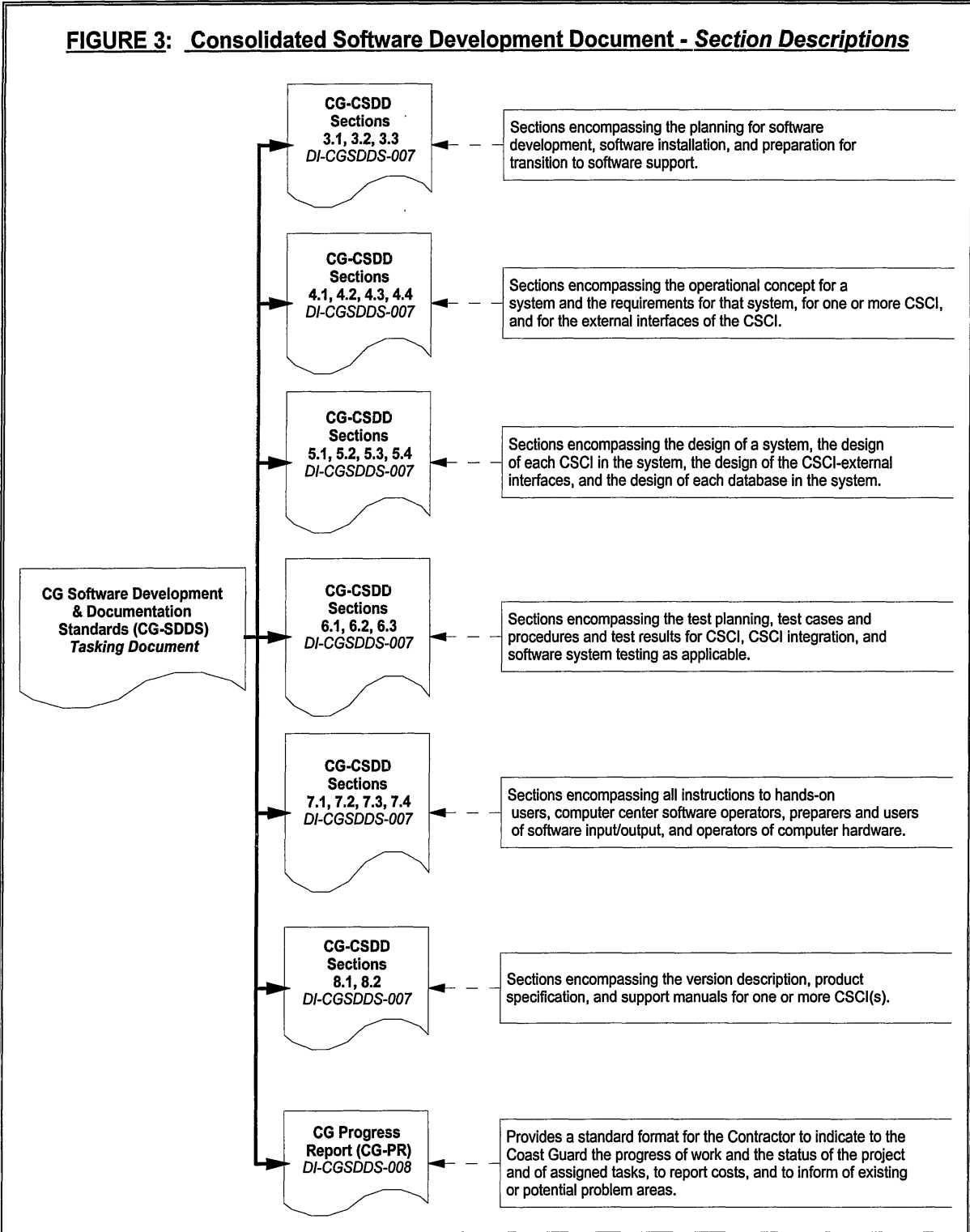


Figure 3: Consolidated Software Development Document-

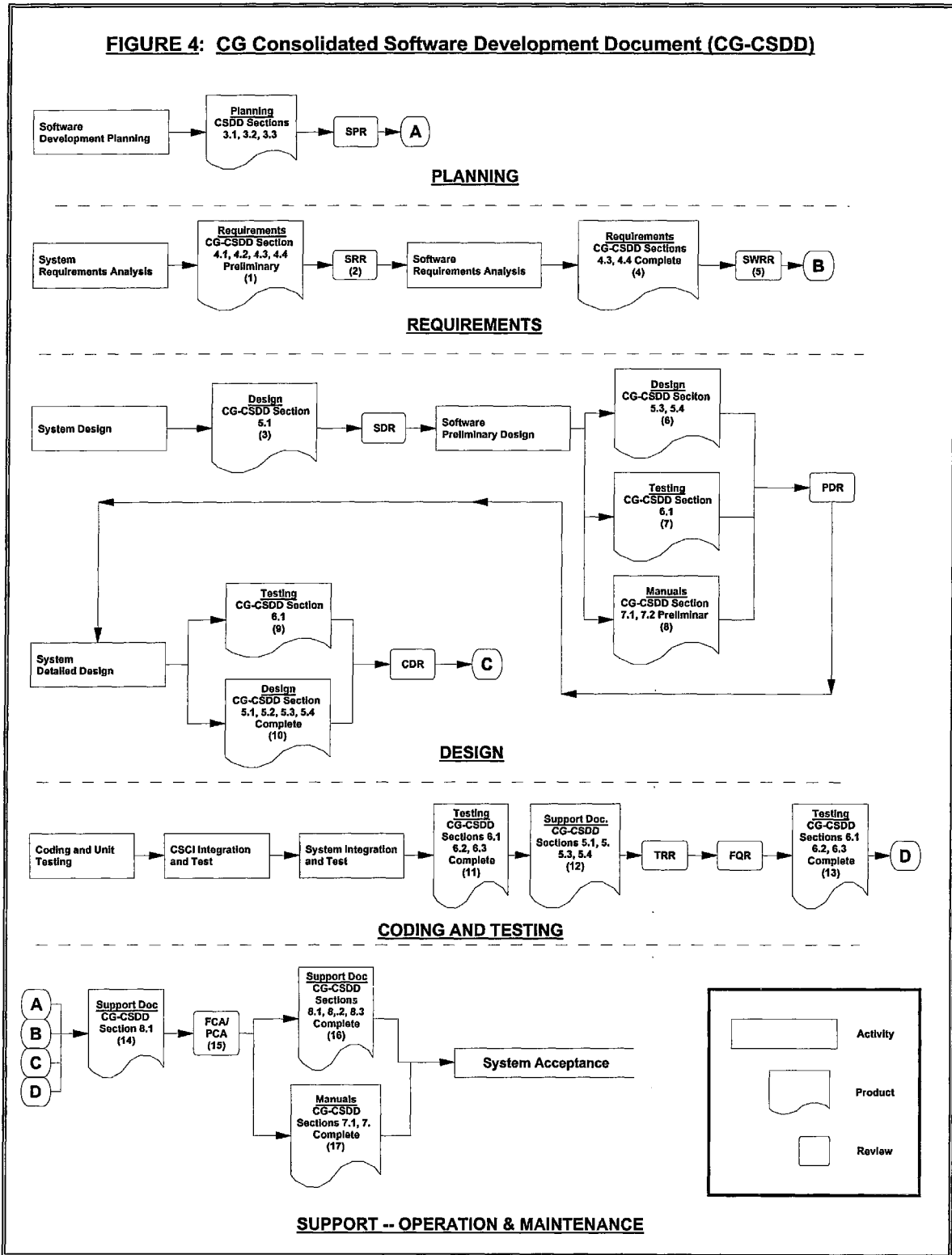


Figure 4: CG Consolidated Software Development Document (CG-CSDD)

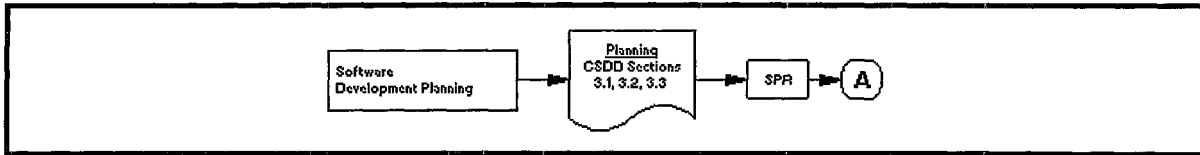
CG Development & Documentation Standards Functional Activities (continued)

Refer to Figure 4

- (1) **System Requirements:** CG-CSDD DID Preliminary Sections 4.1, 4.2, 4.3, 4.4
- (2) **System Requirements Review:** Combine Operational Concepts/System Requirements Review. May also include Software Planning Review: CG-CSDD DID Sections 3.1, 3.2, 3.3
- (3) **Software Design:** CG-CSDD DID Section 5.1
- (4) **Software Requirements:** CG-CSDD DID Completed Sections 4.3, 4.4
- (5) **Software Requirements Review:** CG-CSDD DID Completed Sections 4.1, 4.2, 4.3, 4.4
- (6) **Software Design:** CG-CSDD DID Preliminary Sections 5.3, 5.4
- (7) **Software Testing:** CG-CSDD DID Preliminary Section 6.1
- (8) **Software User/Operator Manual:** CG-CSDD DID Preliminary Sections 7.1, 7.2
- (9) **Software Testing:** CG-CSDD DID Preliminary Section 6.2
- (10) **Software Design:** CG-CSDD DID Completed Sections 5.1, 5.2, 5.3, 5.4
- (11) **Software Testing:** CG-CSDD DID Sections 6.1, 6.2, 6.3
- (12) **Software Support:** CG-CSDD DID Preliminary Section 8.2
- (13) **Software Testing:** CG-CSDD DID Completed Sections 6.1, 6.2, 6.3
- (14) **Software Support:** CG-CSDD DID Section 8.1 (Version Description Documentation)
- (15) **May be held in lieu of, or in addition to, a Software Supportability Review and Software Usability Review**
- (16) **Support Document:** CG-CSDD DID Completed Sections 8.1, 8.2
- (17) **Software User/Operator Manual:** CG-CSDD DID Completed Sections 7.1, 7.2

| Consolidated Software Development Document (CG-CSDD) DID Section | Related CG-SDDS DID Section |
|---|--|
| Section 3.1 Software Development Plan | CG Software Plan (CG-SP) Section 3 |
| Section 3.2 Software Installation Plan | CG Software Plan (CG-SP) Section 4 |
| Section 3.3 Software Support Plan | CG Software Plan (CG-SP) Section 5 |
| Section 4.1 Operational Concept | CG Software Requirements Document (CG-SRD) Section 3.3 |
| Section 4.2 System/Segment Requirements | CG Software Requirements Document (CG-SRD) Section 4 |
| Section 4.3 Software CSCI Requirements | CG Software Requirements Document (CG-SRD) Section 5 |
| Section 4.4 Interface Requirements | CG Software Requirements Document (CG-SRD) Section 6 |
| Section 5.1 System/Segment Design | CG Software Design Document (CG-SDD) Section 3 |
| Section 5.2 CSCI Design | CG Software Design Document (CG-SDD) Section 4 |
| Section 5.3 Interface Design | CG Software Design Document (CG-SDD) Section 5 |
| Section 5.4 Database Design | CG Software Design Document (CG-SDD) Section 6 |
| Section 5.1 Software Test Planning | CG Software Test Document (CG-STD) Section 3 |
| Section 6.2 Software Test Descriptions | CG Software Test Document (CG-STD) Section 4 |
| Section 6.3 Software Test Report | CG Software Test Document (CG-STD) Section 5 |
| Section 7.1 Software User Manual | CG Software User/Operator Manual (CG-SUOM) Section 3 |
| Section 7.2 Computer Center Software Operator Manual | CG Software User/Operator Manual (CG-SUOM) Section 4 |
| Section 7.3 Software Input/Output Manual | CG Software User/Operator Manual (CG-SUOM) Section 5 |
| Section 7.4 Computer System Operator Manual | CG Software User/Operator Manual (CG-SUOM) Section 6 |
| Section 8.1 Version Description | CG Software Support Document (CG-SSD) Section 3 |
| Section 8.2 Software Product Specification | CG Software Support Document (CG-SSD) Section 4 |

Figure 5: Crosswalk of the CG Consolidated Software Development Document (CG-CSDD) DID to Other CG-SDDS DIDs

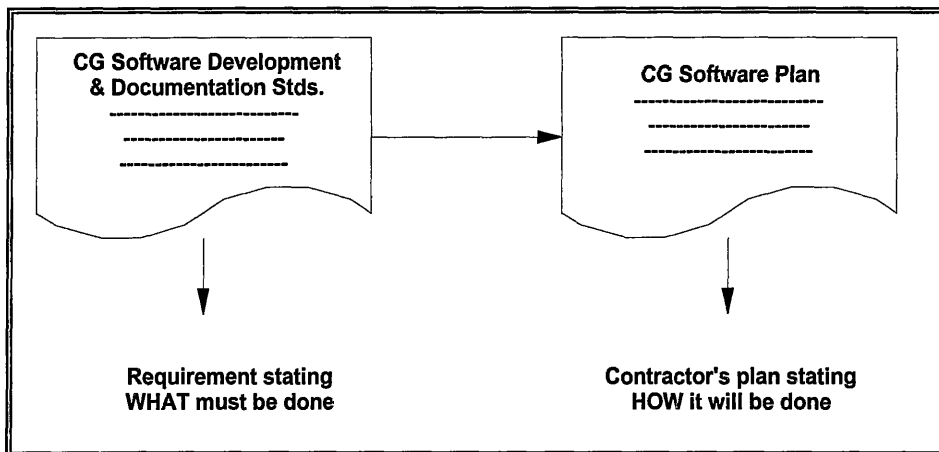


PLANNING PHASE:

Software Development Planning. Coast Guard developed requirements are provided to contractor(s). Based on the Coast Guard's requirements the contractor(s) develops a Software Plan (CG-CSDD DID Sections 3.1, 3.2, and 3.3) that presents the contractor's understanding of the requirement, and the contractor's approach to fulfilling the requirement.

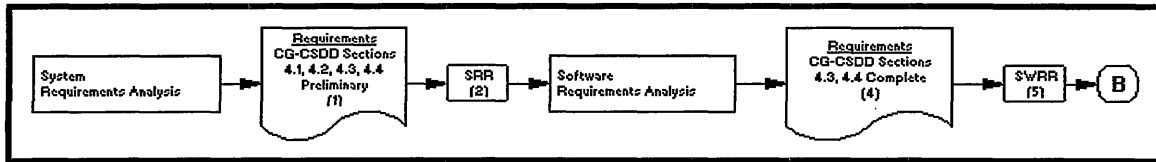
Software Plan -- CG-CSDD DID Sections 3.1, 3.2, and 3.3. These sections encompass the planning for software development, software installation, and preparation for transition to software support. Describes the contractor's plans for conducting a software development effort. Covers new development, modification, reuse, reengineering, maintenance, and all other activities resulting in software products. Provides the Coast Guard insight into, and a tool for monitoring, the processes to be followed for software development, the methods to be used, the approach to be followed for each activity, and project schedules, organization, and resources. Portions of the plan may be bound separately if the approach enhances their usability. Examples include plans for software configuration management and software quality assurance. Software Plan Review (SPR). The SPR is the most important review of the software development effort. It serves as the foundation of understanding between the Coast Guard and the contractor(s) based on the contents of the contractor-developed sections 3.1, 3.2, and 3.3 of the CG-CSDD DID. The SPR will be conducted after the Coast Guard has received and reviewed the initial CG-SP from the contractor for the related software development initiative. A successful SPR is predicated upon the Coast Guard's determination that sections 3.1, 3.2, and 3.3 of the CG-CSDD DID form a satisfactory basis for proceeding into System Requirements Analysis.

Methodology Independence. It is the CG-SDDS's goal to neither encourage nor preclude any particular software development methodology. As shown in the figure below, the standard lays out a set of activities that must be accomplished and defines a set of software products that must be produced, but leaves it to the developer to propose in the Coast Guard Software Plan (CG-SP) how those activities will be performed and how the products will be produced. The Coast Guard has the opportunity to react **to the methods in reviewing** sections 3.1, 3.2, and 3.3 of the CG-CSDD DID.



END PLANNING PHASE:

Using the Consolidated Software Development Document (CG-CSDD) DID



REQUIREMENTS PHASE:

Using the CG Consolidated Software Development Document (CG-CSDD) DID

System Requirements Analysis. The analysis and documentation of system requirements that involves transforming an operational need into a system description, system performance parameters, and a system configuration through the use of an iterative process of analysis, design, trade-off studies, and prototyping.

Software Requirements -- CG-CSDD DID Sections 4.1, 4.2, 4.3, and 4.4.

These sections encompass the operational concept for a system and the requirements for that system, for one or more Computer Software Configuration Items (CSCI), and for the external interfaces of the CSCIs. Specifies the requirements for a system and its supporting CSCI(s) and the methods to be used to ensure that each requirement has been met, and provides guidance for and comprises the following information: Operation Capabilities Description, System/Segment Specification, Software Requirements Specification, and Interface Requirements Specification. (1) **CG-CSDD DID Preliminary Sections 4.1, 4.2, 4.3, and 4.4.**

System Requirements Review (SRR). The object of this review is to ascertain the adequacy of the contractor's efforts in defining the system requirements via sections 4.1, 4.2, 4.3, and 4.4 of the CG-CSDD DID.. The SRR will be conducted after the Coast Guard has received and reviewed the preliminary versions of sections 4.1, 4.2, 4.3, and 4.4 from the contractor for the related software development initiative. A successful SRR is predicated upon the Coast Guard's determination that sections 4.1, 4.2, 4.3, and 4.4 of the CG-CSDD DID form a satisfactory basis for proceeding into preliminary software design. (2) *Combine Operation Concepts/Systems Requirement Review. May also include Software Plan Review: CG-CSDD Sections 3.1, 3.2, and 3.3.*

Software Requirements Analysis. Analyzes the software problem at hand and presents a complete specification of the desired external behavior of the software system to be built. The analysis involves partitioning system requirements into major subsystems and tasks, allocating those subsystems or tasks to software, and transforming these allocated system requirements into a description of software requirements and performance parameters through use of an iterative process of analysis, design, trade-off studies, and prototyping. Also called *software functional description software functional requirements, software requirements specification* by others.

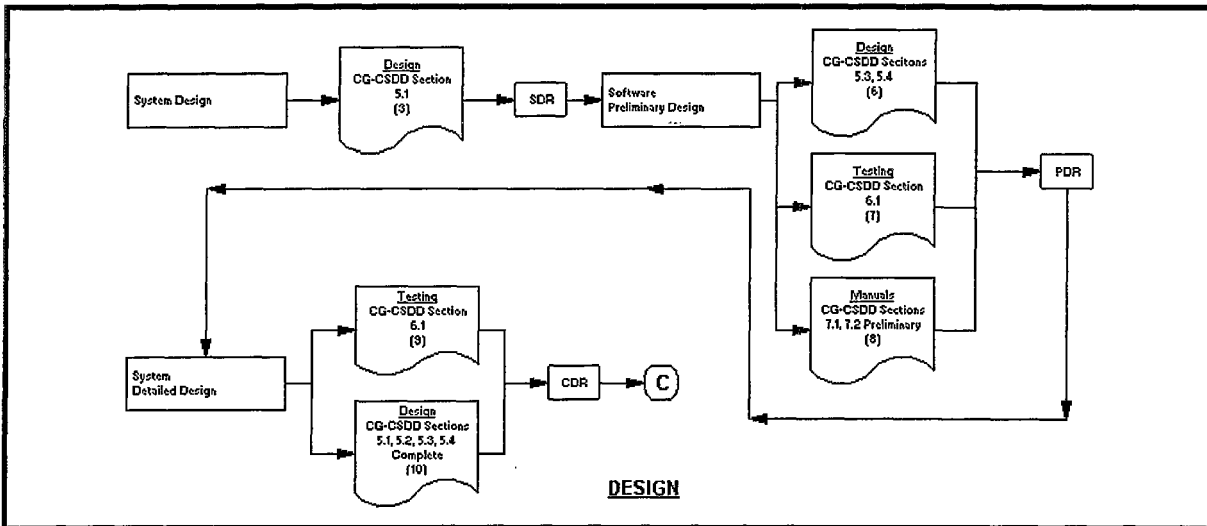
NOTE: The major difference between system requirements analysis and software requirements analysis is that the origin of system requirements lies in user needs while the origin of software requirements lies in the system requirements and/or specification.

Software Requirements -- CG-CSDD DID Sections 4.1, 4.2, 4.3, and 4.4. These sections encompass the operational concept for a system and the requirements for that system, for one or more Computer Software Configuration Items (CSCI), and for the external interfaces of the CSCIs. Specifies the requirements for a system and its supporting CSCI(s) and the methods to be used to ensure that each requirement will be met, and provides guidance for and comprises the following information: Operation Capabilities Description, System/Segment Specification, Software Requirements Specification, and Interface Requirements Specification. (4) **CG-CSDD DID Completed Sections 4.3, and 4.4.**

Software Requirements Review (SWRR). A review of the finalized CSCI requirements and operational concept. The SWRR is conducted when CSCI requirements have been sufficiently defined to evaluate the contractor's responsiveness to and interpretation of the system, segment, or prime item level requirements. A successful SWRR is predicated upon the contracting agency's determination that the *CG Software Requirements Document (CG-CSDD Sections 4.1, 4.2, 4.3, 4.4)* forms a satisfactory basis for proceeding into preliminary software design. 5) **CG-CSDD DID Completed Sections 4.1, 4.2, 4.3, and 4.4.**

END REQUIREMENTS PHASE:

Using the Consolidated Software Development Document (CG-CSDD) DID



DESIGN PHASE:

Using the Consolidated Software Development Document (CG-CSDD) DID

System Design. Analysis and determination of major system components and their purposes. Software and hardware requirements for each of the determined system components is separately specified. The process of defining the hardware and software architecture, components, modules, interfaces, and data for a system to satisfy specified system requirements.

Software Design - CG-CSDD DID Sections 5.1, 5.2, 5.3, and 5.4. These sections encompass the design of a system, the design of each CSCI in the system, the design of the CSCI-external interfaces, and the design of each database in the system. Describes the system-wide design decisions, the system architectural design, and the detailed design needed to implement the software. Provides the Coast Guard visibility into the design and provides information needed for software support. (3) CG-CSDD DID section 5.1.

Software Design Review (SDR) This review shall be conducted to evaluate the optimization, correlation, completeness, and risks associated with the allocated technical requirements as presented in section 5.1 of the CG-CSDD DID. The SDR will be conducted after the Coast Guard has received and reviewed the preliminary version of section 5.1 of the CG-CSDD DID from the contractor for the related software development initiative. A successful SDR is predicated upon the Coast Guard's determination that section 5.1 forms a satisfactory basis for proceeding with the preliminary software design.

Software Preliminary Design. Decomposes the software system into its actual constituent (architectural) components and then iteratively decomposes those components into smaller and smaller subcomponents until the subcomponents located at the leaves of the resulting design tree are small enough for a person to be able to "get his or her arms around it" easily. Each of these modules is documented in terms of its inputs, outputs, and functions; also called *Preliminary Specifications, Preliminary High-level Design, Preliminary Architectural Design, and Preliminary Functional Design.*

Software Design - CG-CSDD DID Sections 5.1, 5.2, 5.3, and 5.4. These sections encompass the design of a system, the design of each CSCI in the system, the design of the CSCI-external interfaces, and the design of each database in the system. Describes the system-wide design decisions, the system architectural design, and the detailed design needed to implement the software. Provides the Coast Guard visibility into the design and provides information needed for software support. 6) CG-CSDD DID **Completed** Sections 5.1, 5.2, 5.3, and 5.4.

DESIGN PHASE: Using the Consolidated Software Development Document (CG-CSDD) DID
(continued)

Software Testing - CG-CSDD DID Sections 6.1, 6.2, and 6.3. These sections encompass the test planning, test cases and procedures, and test results for CSCI, CSCI integration, and software system testing as applicable. Provides a record of the qualification testing performed on a Computer Software Configuration Item (CSCI), a software subsystem, or other related item. Enables the Coast Guard to assess the testing and its results. Sections 6.1, 6.2, and 6.3 provide guidance for and comprise the following information: a Software Test Plan, a Software Test Description, and a Software Test Report. (7) CG-CSDD DID **Preliminary** Section 6.1.

Software User/Operator Manuals - CG-CSDD DID Sections 7.1 and 7.2. These sections encompass all instructions to hands-on users, computer center software operators, preparers and users of software input/output, and operators of computer hardware. These sections provide guidance for and comprise the following information: a Software User Manual, a Computer Center Software Operator Manual, Software Input/Output Manual, Computer System Operator Manual. (8) CG-CSDD DID **Preliminary** Sections 7.1 and 7.21.

Preliminary Design Review (PDR) This review shall be conducted for each configuration item or aggregate of configuration items to:

- (1) evaluate the progress, technical adequacy, and risk resolution (on technical, cost, and schedule basis) of the selected design approach,
- (2) determine its compatibility with performance and engineering speciality requirements of the Hardware Configuration Item (HWCI) development specification,
- (3) evaluate the degree of definition and assess the technical risk associated with the selected development methods/processes, and
- (4) establish the existence and compatibility of the physical and functional interfaces among the configuration item and other items of equipment, facilities, computer software, and personnel.

For CSCIs, this review will focus on:

- (1) the evaluation of the progress, consistency, and technical adequacy of the selected top-level design and test approach,
- (2) compatibility between software requirements and preliminary design, and
- (3) on the preliminary version of the operation and support documents.

The PDR will be conducted after the Coast Guard has received delivery and reviewed the initial CG-SDD from the contractor for the related software development initiative. A successful PDR is predicated upon the Coast Guard's determination that CG-CSDD DID sections:

- * 5.1, 5.2, 5.3, and 5.4 - Software Design,
- * 6.1, 6.2, and 6.3 - Software Testing, and
- * the Preliminary 7.1, 7.2, 7.3, and 7.4 - User/Operator Manuals,

form a satisfactory basis for proceeding into System Detailed Design.

System Detailed Design. Develop design descriptions of each software unit of the CSCI. Define and document algorithms for each module in the design tree that will be realized as code; also called *program design*.

DESIGN PHASE:

Using the Consolidated Software Development Document (CG-CSDD) DID (continued)

Software Testing - CG-CSDD DID Sections 6.1, 6.2, and 6.3. These sections encompass the test planning, test cases and procedures, and test results for CSCI, CSCI integration, and software system testing as applicable. Provides a record of the qualification testing performed on a Computer Software Configuration Item (CSCI), a software subsystem, or other related item. Enables the Coast Guard to assess the testing and its results. Sections 6.1, 6.2, and 6.3 provide guidance for and comprise the following information: a Software Test Plan, a Software Test Description, and a Software Test Report. (9) CG-CSDD DID **Preliminary** 6.1.

Software Design - CG-CSDD DID Sections 5.1, 5.2, 5.3, and 5.4. These sections encompass the design of a system, the design of each CSCI in the system, the design of the CSCI-external interfaces, and the design of each database in the system. Describes the system-wide design decisions, the system architectural design, and the detailed design needed to implement the software. Provides the Coast Guard visibility into the design and provides information needed for software support. (10) CG-CSDD DID **Completed** Sections 5.1, 5.2, 5.3, and 5.4.

Critical Design Review (CDR). This review shall be conducted for each configuration item when detail design is essentially complete. The purpose of this review will be to:

- (1) determine that the detail design of the configuration item under review satisfies the performance and engineering requirements presented in sections 4.1, 4.2, 4.3, 4.4 and sections 5.1, 5.2, 5.3, 5.4,
- (2) determine that the detail design supports compatibility among the configuration item and other items of equipment, facilities, computer software and personnel, and
- (3) assess configuration item risk areas (on a technical, cost, and schedule basis).

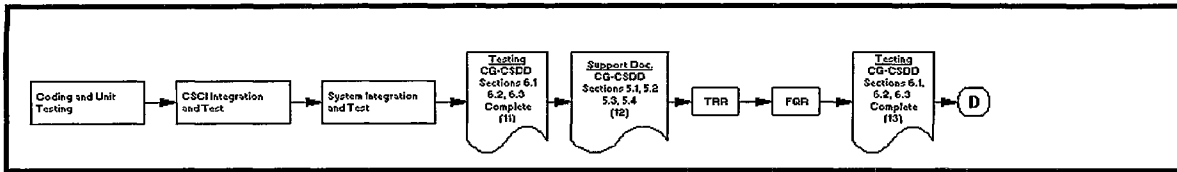
For CSCIs, this review will focus on the determination of the acceptability of the detailed design, performance, and test characteristics of the design solution, and on the adequacy of the operation and support documents:

- * 3.1, 3.2, and 3.3 - Software Plan,
- * 4.1, 4.2, 4.3, and 4.4 - Software Requirements,
- * 5.1, 5.2, 5.3, and 5.4 - Software Design,
- * 6.1, 6.2, and 6.3 - Software Testing,
- * 7.1, 7.2, 7.3, 7.4 - User & Operator Manuals.

A successful CDR is predicated upon the Coast Guard's determination that the completed sections 5.1, 5.2, 5.3 and sections 6.1, 6.2, and 6.3 form a satisfactory basis for proceeding into *Coding and Testing*.

END DESIGN PHASE:

Using the Consolidated Software Development Document (CG-CSDD) DID



CODING AND TESTING PHASE:

Using the Consolidated Software Development Document (CG-CSDD) DID

Coding and Unit Testing. Transforming algorithms defined during the detailed design stage into a computer-understandable language. Check each coded module for the presence of bugs. Unit testing's purpose is to ensure that each as-built module behaves according to its specification defined during detailed design.

CSCI Integration and Testing. Interconnecting set of previously previously tested modules to ensure that the sets behave as well as they did as independently tested modules. Thus, CSCI integration testing's purpose is to ensure that each as-built component behaves according to its specification defined during preliminary design.

System Integration and Test. Check that the entire (i.e., fully integrated) software system embedded in its actual hardware environment behaves according to sections 4.1, 4.2, 4.3, and 4.4 of the CG-CSDD DID.

Software Testing - CG-CSDD DID Sections 6.1, 6.2, 6.3, and 6.4. These sections encompass the test planning, test cases and procedures, and test results for CSCI, CSCI integration, and software system testing as applicable. Provides a record of the qualification testing performed on a Computer Software Configuration Item (CSCI), a software subsystem, or other related item. Enables the Coast Guard to assess the testing and its results. Sections 6.1, 6.2, and 6.3 of the CG-CSDD provide guidance for and comprise the following information: a Software Test Plan, a Software Test Description, and a Software Test Report. (11) *CG-CSDD DID Sections 6.1, 6.2, and 6.3.*

Software Support - CG-CSDD DID Sections 8.1 and 8.2. These sections encompass the version description, product specification, and support manuals for one or more CSCI(s). (12) *CG-CSDD DID Preliminary Sections 8.1 and 8.2.*

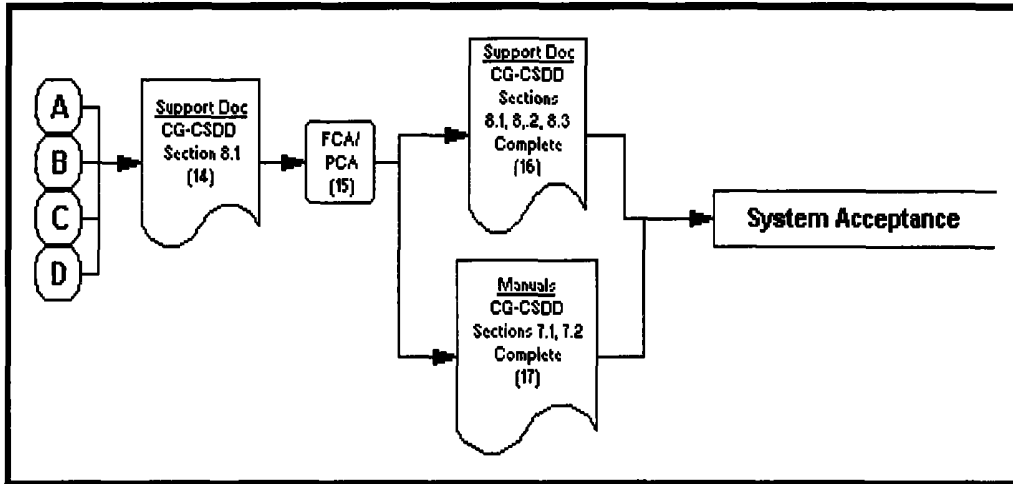
Test Readiness Review (TRR). A review conducted for each CSCI to determine whether the software test procedures are complete. Software test procedures are evaluated for compliance with the software test plans and descriptions (CG-CSDD DID sections 6.1, 6.2, and 6.3) and for adequacy in accomplishing test requirements. At TRR, the Coast Guard also reviews the results of software testing and any updates to support documentation.

Formal Qualification Review (FQR). The test, inspection, or analytical process by which a group of configuration items comprising the system are verified to have met specific Coast Guard contractual performance requirements (specifications or equivalent).

Software Testing - CG-CSDD DID Sections 6.1, 6.2, 6.3, and 6.4. These sections encompass the test planning, test cases and procedures, and test results for CSCI, CSCI integration, and software system testing as applicable. Provides a record of the qualification testing performed on a Computer Software Configuration Item (CSCI), a software subsystem, or other related item. Enables the Coast Guard to assess the testing and its results. Sections 6.1, 6.2, and 6.3 of the CG-CSDD provide guidance for and comprise the following information: a Software Test Plan, a Software Test Description, and a Software Test Report. (13) *CG-CSDD DID Completed Sections 6.1, 6.2, and 6.3.*

END CODING AND TESTING PHASE:

Using the Consolidated Software Development Document (CG-CSDD) DI



SUPPORT -- OPERATION & MAINTENANCE:

Using the Consolidated Software Development Document (CG-CSDD) DID

All outputs from the previous phases flow into the Support -- Operation & Maintenance Phase. The phase outlines the final steps before system acceptance and ensures that all support documentation has been properly delivered. This phase also supports operations and maintenance by ensuring that all baseline specifications (documentation) will be updated when Change Requests occur.

Software Support - CG-CSDD DID Sections 8.1 and 8.2. These sections encompass the version description, product specification, and support manuals for one or more CSCI(s). (14) *CG-CSDD DID Section 8.1 (Version Description Documentation).*

Functional Configuration Audit (FCA) / Physical Configuration Audit (PCA). FCA - A formal audit to validate that the development of a configuration item has been completed satisfactorily and that the configuration item has achieved the performance and functional characteristics specified in the functional or allocated configuration identification. In addition, the completed operation and support documents shall be reviewed.

PCA -A technical examination of a designated configuration item to verify that the configuration item "As Built" conforms to the technical documentation which defines the configuration item. (15) *May be held in lieu of, or in addition to, a Software Supportability Review and Software Usability Review.*

Software Support - CG-CSDD DID Sections 8.1 and 8.2. These sections encompass the version description, product specification, and support manuals for one or more CSCI(s). (16) *CG-CSDD DID Completed Sections 8.1 and 8.2.*

Software User/Operator Manual - CG-CSDD Sections 7.1, 7.2, 7.3, and 7.4. These sections encompass all instructions to hands-on users, computer center software operators, preparers and users of software input/output, and operators of computer hardware. Sections 7.1, 7.2, 7.3, and 7.4 provide guidance for and comprise the following information: a Software User Manual, a Computer Center Software Operator Manual, Software Input/Output Manual, Computer Systems Operator Manual. (17) *CG-CSDD DID Completed Sections 7.1, 7.2, 7.3, and 7.4. **System Acceptance.***

END SUPPORT -- OPERATION & MAINTENANCE:

Using the Consolidated Software Development Document (CG-CSDD) DID

References:

DoD Harmonization Working Group.

Military Standard 498. USA: The Department of Defense, 1994.

DoD Harmonization Working Group.

Military Standard 498; Handbook Volume 1. USA: The Department of Defense, 1994.

Davis, Alan M. Software Requirements; Objects, Functions, & States.

USA: Prentice-Hall, 1993.

R. H. Thayer and M. Dorfman.

System and Software Requirements Engineering. USA: IEEE Computer Society Press, 1990.

SAMPLE GENERIC STATEMENT OF WORK

Referencing the
COAST GUARD CONSOLIDATED SOFTWARE DEVELOPMENT DOCUMENT
APPENDIX A

SECTION C

Table of Contents

| | |
|--|------------|
| C.1 GENERAL | C-1 |
| C.1.1 Scope | C-1 |
| C.1.2 Background | C-1 |
| C.1.3 Contractor Personnel | C-2 |
| C.1.4 Quality | C-2 |
| C.2 DEFINITIONS | C-2 |
| C.3 GOVERNMENT FURNISHED PROPERTY/INFORMATION | C-2 |
| C.4 CONTRACTOR FURNISHED ITEMS | C-2 |
| C.5 SPECIFIC TASKS | C-3 |
| C.5.1 Conversion of the CG Mission Essential Application (CGMEA), CLIN 0001 | C-4 |
| C.5.1.1 Software Development Planning, CLIN 0001A | C-4 |
| C.5.1.2 Systems Requirements Analysis, CLIN 0001B | C-4 |
| C.5.1.3 System Design, CLIN 0001C | C-5 |
| C.5.1.4 Software Requirements Analysis, CLIN 0001D | C-5 |
| C.5.1.5 Preliminary Software Design, CLIN 0001E | C-5 |
| C.5.1.6 Detailed Software Design, CLIN 0001F | C-6 |
| C.5.1.7 Code Development and Unit Testing, CLIN 0001G | C-7 |
| C.5.1.8 CGMEA CSCI Integration Testing, CLIN 0001H | C-7 |
| C.5.1.9 CGMEA System Tuning, CLIN 0001I | C-7 |
| C.5.1.10 System Integration and Testing, Test Readiness Review (TRR), CLIN 0001J | C-7 |
| C.5.1.11 Formal Qualification Testing/ Acceptance, CLIN 0001K | C-7 |
| C.5.2 Data for the CG Mission Essential Application (CGMEA) Conversion, CLIN 0002 | C-8 |
| C.5.2.1 Software Planning, CLIN 0002A | C-8 |
| C.5.2.2 Software Requirements Documentation, CLIN 0002B | C-9 |
| C.5.2.3 Software Design Documentation, CLIN 0002C | C-9 |

| | | |
|---------|--|-----|
| C.5.2.4 | Software Test Documentation, CLIN 0002D | C-9 |
| C.5.2.5 | Software Support Documentation, CLIN 0002E | C-9 |
| C.5.2.6 | Computer Software: Converted CG Mission Essential Application (CGMEA), CLIN 0002F | C-9 |

Table of Contents

| | |
|---|-------------|
| C.5.3 CG Mission Essential Application (CGMEA) Legacy data Migration, CLIN 0003 | C-9 |
| C.5.3.1 Migration Analysis, CLIN 0003A | C-10 |
| C.5.3.2 Requirements Analysis, CLIN 0003B | C-10 |
| C.5.3.3 Design/Development of Legacy Data Migration Utility, CLIN 0003C | C-10 |
| C.5.3.4 Code and Unit Test: Data Migration Utility, CLIN 0003D | C-10 |
| C.5.3.5 CGMEA Legacy Data Migration Process, CLIN 0003E | C-11 |
| C.5.3.6 CG Mission Essential Application Legacy Data Migration, CLIN 0003F (optional) | C-11 |
| C.5.4 Data for CG Mission Essential Application (CGMEA) Legacy Data Migration, CLIN 0004 | C-11 |
| C.5.4.1 Technical Report on Data Migration Risk Analysis, CLIN 0004A | C-11 |
| C.5.4.2 Legacy Data Migration Software Requirements Document, CLIN 0004B | C-11 |
| C.5.4.3 Legacy Data Migration Software Design Document (SDD), CLIN 0004C | C-11 |
| C.5.4.4 Legacy Data Migration Process Guide, Software User/Operator Manual (SUOM), CLIN 0004D | C-11 |
| C.5.4.5 CGMEA Legacy Data Migration Utility CSCI, CLIN 0004E | C-12 |
| C.5.5 Training, CLIN 0005 | C-12 |
| C.5.5.1 End User Training, CLIN 0005A | C-12 |
| C.5.5.2 Develop CGMEA Software User and Operators Procedures, CLIN 0005B | C-12 |
| C.5.6 Data for the CG Mission Essential Application (CGMEA) Training, CLIN 0006 | C-12 |
| C.5.6.1 CGMEA User/Operator Manual (SUOM), CLIN 0006A | C-12 |
| C.5.7 Contract Management, CLIN 0007 | C-13 |

| | | |
|---------|--|------|
| C.5.7.1 | Management Process Documentation; Requirements of Software Planning, CLIN 0007A | C-13 |
| C.5.7.2 | Software Quality Assurance (QA), CLIN 0007B | C-13 |
| C.5.7.3 | Software Configuration Management (SCM), CLIN 0007C | C-13 |
| C.5.7.4 | Technical Interchange Meetings, Reviews and Audits, CLIN 0007D | C-14 |

Table of Contents

| | | |
|--------------|---|-------------|
| C.5.7 | Contract Management, CLIN 0007 (Cont.) | |
| C.5.7.5 | Project Status Reporting, CLIN 0007E | C-14 |
| C.5.7.6 | Metrics, CLIN 0007F | C-14 |
| C.5.7.7 | Government Directed Travel, CLIN 0007G | C-15 |
| C.5.8 | Data for Project Management, CLIN 0008 | C-15 |
| C.5.8.1 | Monthly Progress Report, CLIN 0008A | C-15 |
| C.5.8.2 | Technical Interchange Meeting (TIM) and Review Agenda, CLIN 0008B | C-15 |
| C.5.8.3 | Minutes from TIM, Reviews, and Audits. CLIN 0008C | C-15 |
| C.6 | APPLICABLE REFERENCE DOCUMENTS | C-16 |
| C.7 | CONTRACTOR PERSONNEL REQUIREMENTS | C-16 |
| C.7.1 | Key Personnel | C-16 |
| C.7.2 | Personnel Qualifications | C-17 |
| C.7.2.1 | Contractor Program Manager | C-17 |
| C.7.2.2 | Senior Computer System Analyst | C-18 |
| C.7.2.3 | Quality Assurance Manager | C-18 |
| C.7.2.4 | Computer Systems Analyst | C-19 |
| C.7.2.5 | Senior Software Engineer | C-20 |
| C.7.2.6 | Application Programmer/Software Engineer | C-21 |
| C.7.2.7 | Database Management System Specialist | C-22 |
| C.7.2.8 | Documentation Specialist | C-22 |
| C.8 | CONTRACTOR SECURITY REQUIREMENTS | C-23 |
| C.8.1 | General Security | C-23 |
| C.8.2 | Administrative Security - Unclassified Systems | C-24 |
| C.8.3 | Malicious Software | C-25 |
| C.9 | TRANSITION PLANNING AND SUPPORT SERVICES | C-25 |
| C.10 | NOTES | C-25 |
| C.11 | PERFORMANCE REQUIREMENTS SUMMARY | C-26 |

PART I, SECTION C
DESCRIPTION/SPECIFICATION/WORKSTATEMENT

C.1 GENERAL

C.1.1 Scope

- 1.1.1 This Statement of Work (SOW) defines the effort required to perform software design, development, testing, documentation, and training for the conversion and integration of the Coast Guard's CG Mission Essential Application (CGMEA), and the migration of legacy CGMEA data from the current USCG Standard Workstation Environment (CGSWII) to that of the next generation Standard Workstation Environment (CGSWIII).
- 1.1.2 The Coast Guard defines Software Conversion as the transformation, without functional change, of computer programs and data elements to permit their use on a replacement or changed hardware/software environment. Thus, this effort includes those tasks necessary to accomplish that conversion.
- 1.1.3 Other objectives within this scope include full documentation of the converted application to adopted standards, an analysis of legacy data migration, the development of data conversion tools and utilities to effect that migration, and training of a select set of management and end user domain experts on the newly delivered system.

C.1.2 Background

- 1.2.1 The U.S. Coast Guard is in the process of procuring a new client/server computing environment under the Coast Guard's Standard Workstation III (CGSWIII) procurement. The CGSWIII procurement complies with Federal Information Processing Standards (FIPS) and includes Intel based workstation and server hardware, and the Windows NT operating system environment. The software environment includes the operating system on all workstation and server platforms, the Informix SE Online Relational Database Management System (RDBMS), the Windows graphical user

interface, Powerbuilder applications development tools and C/C++, Pascal and Ada language tools. As a result, the Coast Guard has identified the CG Mission Essential Application (CGMEA) as one of a number of Mission Essential Applications (MEAs) requiring conversion from the current proprietary operating environment to that of CGSWIII.

- 1.2.2 Documentation and software development processes in support of conversion and integration are to be guided by the Coast Guard Software Development and Documentation Standard (SDDS). This comprehensive standard is based on MIL-STD-498, tailored for the Coast Guard. A full list of referenced documents and provisions as to how to obtain those documents is provided in Section J (Attachment J-1) of this contract.

C.1.3 Contractor Personnel

- a. The Contractor shall provide and maintain personnel qualified to perform the tasks necessary to provide the items and services required by this Contract.
- b. The majority of this effort is to be performed at the Contractor's own facilities.
- c. Additional Contractor Personnel requirements are presented in this Work Statement in Section C.7, Contractor Personnel Requirements.

C.1.4 Quality

- a. The Contractor is responsible for delivering items and services which meet the requirements of this Work Statement.
- b. Additional quality requirements are presented as part of the Contract Management Task.

C.2 DEFINITIONS

Attachment J-2 contains a list of acronyms and definitions for terms used in this Work Statement.

C.3 GOVERNMENT FURNISHED PROPERTY/INFORMATION

- a. Attachment J-3 contains the listing of Government Furnished Property and Information for this Work Statement.
- b. The Contractor shall manage, maintain, and provide accurate accounting for all Government Furnished Property (GFP) and Government Furnished Information (GFI).

C.4 CONTRACTOR FURNISHED ITEMS

Except for property or information specifically stated to be Government furnished in Section C.3, the contractor shall furnish everything required to perform the work described by this Work Statement. Contractor furnished items and services shall be in accordance with the references and standards listed in this Work Statement and to the technical specifications and functional requirements contained in Section J (Attachments).

C.5 SPECIFIC TASKS

The work required by this contract shall be performed in accordance with the listed references, standards, technical specifications and functional requirements contained in Section J (Attachments). A listing of all attachments is provided in Section C.10 Notes, of this work statement. Examples of attachments include the CGMEA Performance Specification (J-4), the source and target environments of the CGMEA application (J-5) and the Development Environment Specification (J-6). The contractor shall ensure that the system delivered is developed to the terms of this contract. Application development and documentation is to be performed in accordance with the Coast Guard Software Development and Documentation Standards (SDDS) and the Coast Guard Software Engineering Guidelines. The application development process shall be documented in accordance with the *Consolidated Software Development Document (CSDD)*, Exhibit HH of the SDDS). The CSDD shall be used as the primary documentation deliverable. Extensive reference to processes, activities, data items and deliverables as

defined in the SDDS are made in this work statement. These specific references are identified with *italicized* type. Technical data to be furnished hereunder shall be prepared in accordance and as defined in the CSDD and Section J (Attachment J-7) to this contract, Contract Data Requirements List (CDRL).

In planning and executing the tasks contained in this DO, the Contractor shall ensure conformance to the following broad requirements:

- a. Ensure transformation of the application into the CGSWIII environment without functional change by providing equivalent functionality, inputs, and outputs.
- b. Maintain essentially the same form and content screens with the addition of GUI functionality.
- c. Ensure converted application produces identical results through stringent testing to include processing comparisons with the existing application.
- d. Emulation of the current CGSWII application environment on CGSWIII is not to be construed as meeting the requirements of this SOW.
- e. Do not construe this conversion and integration effort as requiring a line-for-line conversion.
- f. Ensure preservation and improvement of existing processing efficiencies while improving portability and maintainability of the converted application.
- g. The Contractor shall employ structured analysis and design techniques. Where not otherwise specified, the Yourdon/DeMarco methodology shall be used for modeling the process of information flow and content.

The contractor shall furnish the supplies and services set forth in Section B as follows:

**C.5.1 Conversion of the CG Mission Essential Application (CGMEA),
CLIN 0001**

The Contractor shall develop, document, test, and integrate the CG Mission Essential Application in accordance with the references and standards listed and to the technical specifications and functional requirements contained in Section J (Attachments J-4 through J-6). The Contractor shall ensure that the correct version of commercial off-the-shelf (COTS) software, as specified in Section J (Attachment J-6), are used for all development, test, and integration activities involving COTS. Specific tasks are as follows:

C.5.1.1 Software Development Planning, CLIN 0001A

The Contractor shall implement a software development process and document the process in the CSDD. The development of the *Plans* section will be guided by the listed references, standards, technical specifications and functional requirements contained in Section J (Attachments). The *Software Development, Installation, and Support Plan* sections shall be reviewed at a *Software Plan Review*.

C.5.1.2 System Requirements Analysis, CLIN 0001B

The Contractor shall perform a *System Requirements Analysis* of the CG Mission Essential Application application specified in Attachment J-4. Activities shall include, but are not limited to, review of the CG Mission Essential Application technical specifications and functional requirements, review of supporting application documentation (furnished as GFI), interviews with Government designated domain experts and system users, and examination of the existing CGMEA application in its current environment (CGSWII). The contractor shall document the results of the requirements analysis in the *Concept and Requirements* section of the CSDD. CGMEA business processes shall be documented in accordance with IDEF0 process modeling practices as specified in FIPS Pub 183. The information shall include a preliminary set of qualification requirements. These sections of the CSDD shall be reviewed at a *Systems Requirements Review*.

The Contractor shall update the *CSDD* based on the results of the review.

C.5.1.3 System Design, CLIN 0001C

Following completion of the *Systems Requirements Review*, the Contractor shall develop a *System/Segment Design*. The proposed design shall be documented in the *Design* section of the *CSDD*. This document shall be reviewed at a formal *System Design Review*.

C.5.1.4 Software Requirements Analysis, CLIN 0001D

The Contractor shall perform software requirements analysis for the CGMEA Computer Software Configuration Items (CSCI). The Contractor shall document the complete set of software requirements in the *System/Segment, CSCI, and Interface Requirements* section of the *CSDD*. The *CSDD* shall include a complete set of qualification requirements. These requirements shall be reviewed at a *Software Requirements Review*. The Contractor shall update the *CSDD* based on the results of that review.

C.5.1.5 Preliminary Software Design, CLIN 0001E

The Contractor shall perform preliminary Architecture, Interface and Database software design activities for the CGMEA application. These activities are to be documented in the *Design* section of the *CSDD* in accordance with the approved *Software Planning* sections. During the Preliminary Software Design, the Contractor shall initiate development of a *Software Test Plan and Software User/Operator Manual (SUOM)*. The Contractor shall establish and maintain Software Development Files (SDFs) which shall be subject to review by the Government. The draft *CSDD* sections on *Software Development Planning, Software Testing*, and the Preliminary draft of the *User/Operator Manual* are subject to formal reviews.

- C.5.1.5.1 Architecture. The Contractor shall develop a preliminary software design and document this information in the *CSDD*. The Contractor shall

design the CGMEA application to conform to the Intermediate Layered Architecture as described in the Coast Guard Software Engineering Guidelines.

- C.5.1.5.2 Interface. The Contractor shall develop an interface design in support of the functional requirements specified in the *CSDD*. The Contractor shall document the design in the *CSDD*.
- C.5.1.5.3 Database. The Contractor shall develop a logical database design in support of the functional requirements specified in the approved *Concept and Requirements* section of the *CSDD*. The database design shall contain all data entities and relationships necessary to support and maintain the new software and existing data elements. The logical database design shall be compliant with IDEF1X information modeling practices as specified in FIPS Pub 184 and shall be decomposed to third normal form (3NF). Data element naming conventions shall be in compliance with the Coast Guard Data Element Naming Standards as described in COMDTINST M5230.42(series).
- C.5.1.5.4 Software Test Preparation and Software User/Operators Manual (SUOM) Development. During Preliminary Software Design, the Contractor shall develop a Software Test Plan and document this information in the *Software Testing* section of the *CSDD*. In addition, the Contractor shall initiate development of the *SUOM*.

C.5.1.6 Detailed Software Design, CLIN 0001F

Upon review and approval of the preliminary design(s), the Contractor shall proceed with detailed design activities as described below. The Contractor shall update the *System/Segment Design* to reflect any changes resulting from review of this documentation by the Government. During detailed design, the Contractor shall develop test cases and shall document them in the *CSDD*.

- C.5.1.6.1 Architecture. Upon review and approval of the preliminary design, the Contractor shall develop a detailed system architectural design and document this information in the *CSCI Design* portion of the *CSDD*.
- C.5.1.6.2 Interface. Upon review and approval of the preliminary interface design, the Contractor shall develop a detailed interface design and document this information in the *Interface Design* portion of the *CSDD*.
- C.5.1.6.3 Database. Upon review and approval of the logical database design (3NF), the Contractor shall develop a key-based, fully attributed data model; physical design; and code table specifications and shall document this information in the *Database Design* portion of the *CSDD*.

C.5.1.7 Code Development and Unit Testing, CLIN 0001G

The Contractor shall perform code development and unit testing on the CGMEA CSCI in accordance with the *Software Planning, Design, and Software Testing* sections of the *Consolidated Software Development Document (CSDD)*. The Contractor shall document the results of all testing in CSCI Software Development Folders.

C.5.1.8 CGMEA CSCI Integration Testing, CLIN 0001H

The Contractor shall perform integration and testing on the CGMEA CSCI in accordance with the approved *Software Planning, Design, and Software Testing* Sections of the *CSDD*. The Contractor shall develop test procedures and document plans for testing the system software in the *CSDD*. The Contractor shall update the *CSDD* as necessary to reflect changes in test procedures and document test results in the CGMEA CSCI Software Development Folders.

C.5.1.9 CGMEA System Tuning, CLIN 0001I

The Contractor shall tune the converted application to optimize its performance in the target CGSWIII

environment. Any modifications made to the application to optimize its performance in the target environment shall be documented in the CSDD.

C.5.1.10 System Integration and Testing, Test Readiness Review (TRR), CLIN 0001J

The Contractor shall draft and deliver the *Software Test Planning and Test Description* portions of the CSDD prior to convening a *Test Readiness Review (TRR)*. The TRR shall determine the Contractor's readiness for Formal Qualification Testing (FQT).

C.5.1.11 Formal Qualification Testing/Acceptance, CLIN 0001K

- a. The Contractor shall participate in Formal Qualification Testing (FQT) of each developed CSCI (including the Legacy Data Migration Software developed under CLIN 0003), in accordance with the approved *Software Planning, Software Test Planning, and the Software Test Description portions of the CSDD*. The Contractor shall have received in writing, approvals for these sections and results of the *Test Readiness Review* prior to the commencement of FQT.
- b. Prior to commencement of FQT, the Coast Guard Contracting Officer will designate in writing Government personnel responsible for conducting FQT. This document will specify a Government Spokesperson as the single point of contact for the Government for matters regarding FQT and will be provided to the contractor in advance of FQT.

C.5.1.11.1 Formal Qualification Testing (FQT).

The Contractor shall conduct FQT utilizing procedures documented in the CSDD. The Contractor shall update the CSDD to reflect any changes to tests conducted during FQT. The Contractor shall document the results of this testing in the Software Test Report Section of the CSDD. The Contractor shall make necessary revisions to application code based on the results of FQT and

document those changes in the *CSDD*. The Contractor shall document the "as built" baseline *Software Product Specification* and identify the exact version of the delivered software *Version Description* in the *CSDD*. The Government reserves the right to convene a *Software Supportability Review* to determine the adequacy of the documentation to support the software.

C.5.1.11.2 Acceptance Criteria.

The purpose of FQT is to verify and validate conformance of the converted CG Mission Essential Application software to the requirements specified in the approved sections of the *CSDD*. Acceptance Criteria are specified in Section E of this contract.

C.5.2 Data for the CG Mission Essential Application (CGMEA) Conversion, CLIN 0002

Technical data to be furnished hereunder shall be prepared in accordance with the supporting *Consolidated Software Development Document (CSDD)* Data Item Description as defined in the CG-SDDS, and Section J (Attachment J-7), Contract Data Requirements List (CDRL).

C.5.2.1 Software Development Planning Documentation, CLIN 0002A [CDRL A001]

This Data Item describes format, contents and preparation instructions for incorporation into the *plans* section (3.0) of the *CSDD* of data generated in support of software development planning.

C.5.2.2 Software Requirements Documentation, CLIN 0002B [CDRL A001]

This Data Item describes format, contents and preparation instructions for incorporation into the *Operational Concept/Requirements* section (4.0) of the *CSDD* of data generated in support of system and software requirements analysis.

C.5.2.3 Software Design Documentation, CLIN 0002C [CDRL A001]

This Data Item describes format, contents and preparation instructions for incorporation into the *Design* section (5.0) of the *CSDD* of data generated in support of system and software design.

C.5.2.4 Software Test Documentation, CLIN 0002D [CDRL A001]

This Data Item describes format, contents and preparation instructions for incorporation into the *Software Testing* section (7.0) of the *CSDD* of data generated in support of code, unit, integration and system testing.

C.5.2.5 Software Support Documentation, CLIN 0002E [CDRL A001]

This Data Item describes format, contents and preparation instructions for incorporation into the *Software User/Operators Manual* and the *Support Information* section (8.0) of the *CSDD* of data generated in support of delivered software CSCIs.

C.5.2.6 Computer Software: Converted CG Mission Essential Application (CGMEA), CLIN 0002F [CDRL A002]

The delivered CG Mission Essential Application CSCI shall include source and executable code and any support files necessary to compile, link, or execute the software. The CSCI shall be delivered on magnetic media compatible with CGSWIII hardware and software as specified in Section J (Attachment J-5, Application Environment Specification). Any subsequent delivery shall include a summary report which details any software changes from the previous delivery.

C.5.3 CG Mission Essential Application (CGMEA) Legacy Data Migration, CLIN 0003

Under Government direction, perform an analysis on CGMEA legacy data. Develop, test, and document a data migration process and required automated data migration utilities, and perform CGMEA legacy data migration as specified below.

C.5.3.1 Data Migration Analysis, CLIN 0003A

The Contractor shall perform an analysis of the CGMEA legacy data conversion requirements. The Contractor shall examine the magnitude, integrity and complexity of the existing legacy database(s) to ascertain impact and risks associated with a fully automated conversion, reentry of data, or no conversion. The Contractor shall consider such requirements as mapping routines and integrity checks and the possibility of different default data types in the analysis. The analysis shall be summarized in a Pro/Con analysis followed by recommendations and justification along with any and all cost estimates. The Contractor shall document the results of the analysis in a Technical Report.

C.5.3.2 Requirements Analysis, CLIN 0003B

The Contractor shall perform System Requirements Analysis activities for the CGMEA Data Migration Utility CSCI. The Contractor shall document a complete set of software requirements in the *CSDD*. The *System/Segment Requirements* shall be reviewed at the *Software Requirements Review (SWRR)*. The Contractor shall update these requirements based on the results of the review.

C.5.3.3 Design/Development of Legacy Data Migration Utility, CLIN 0003C

The Contractor shall design and develop an automated data migration utility for use at multiple conversion sites. The Contractor shall analyze existing data structures, define the transformation algorithms to be used to convert the existing data elements to the new structure, define the resulting data elements, and design the software required to perform the transformations and populate the target database(s) and

files. The Contractor shall document this information in the *Design* Sections of the *CSDD*.

C.5.3.4 Code and Unit Test: Data Migration Utility, CLIN 0003D

The Contractor shall perform code and unit testing on the developed data migration utility CSCI in accordance with the appropriate sections of the *CSDD* as approved by the Government. This CLIN is subject to FQT IAW Para 5.1.10. The Contractor shall document the results of all testing in the CSCI Software Development Folders.

C.5.3.5 CGMEA Legacy Data Migration Process, CLIN 0003E

The Contractor shall develop a detailed process for migration of application legacy database(s) and files from the application's current environment to the new application environment. The Contractor shall develop detailed instructions for use by Coast Guard personnel detailing step-by-step procedures for accomplishing data migration from the CGMEA source environment to that of the converted CGMEA applications environment. The Contractor shall document this process in the *Software/User Operator Manual (SUOM)*. The migration process shall be reviewed at the *Software Requirements Review (SWRR)*. The Contractor shall update the SUOM based on the results of the review.

C.5.3.6 CG Mission Essential Application Legacy Data Migration, CLIN 003F (OPTIONAL)

The Contractor shall perform CGMEA legacy data migration, as directed by the Government, at designated Government sites as specified in Section J (Attachment J-8). This data migration shall be performed in accordance with the approved migration process developed and documented in accordance with CLIN 0003E.

C.5.4 Data for CG Mission Essential Application (CGMEA) Legacy Data Migration, CLIN 0004

Technical data to be furnished hereunder shall be prepared in accordance with the supporting *Consolidated Software Development Document (CSDD)* Data Item Description as defined in the CG-SDDS, and Section J (Attachment J-7), Contract Data Requirements List (CDRL).

- C.5.4.1 Technical Report on Data Migration Analysis, CLIN 0004A [CDRL A004]**
- C.5.4.2 Legacy Data Migration Software Requirements Documentation (Section 4, Incorporate into CLIN 0002B), CLIN 0004B [CDRL A001]**
- C.5.4.3 Legacy Data Migration Software Design Documentation (Section 5, Incorporate into CLIN 0002C), CLIN 0004C [CDRL A001]**
- C.5.4.4 Legacy Data Migration Process Guide, Software User/Operator Manual (SUOM) (Incorporate into CLIN 0006A), CLIN 0004D [CDRL A003]**
- C.5.4.5 CGMEA Legacy Data Migration Utility CSCI, CLIN 0004E [CDRL A005]**

The delivered CG Mission Essential Application Legacy Data Migration Utility CSCI shall include source and executable code and any support files necessary to compile, link, or execute the software. The CSCI shall be delivered on magnetic media compatible with CGSWII and/or CGSWIII hardware and software as appropriate and as specified in Section J (Attachment J-5), Application Environment Specification. Any subsequent delivery shall include a summary report which details any software changes from the previous delivery.

C.5.5 Training, CLIN 0005

The Contractor shall provide training and develop training material as specified in Section J (Attachment 9) of this Statement of Work (SOW). The Contractor developed training material will be incorporated as an appendix to the *Software User and Operator Manual (SUOM)*.

C.5.5.1 End User Training, CLIN 0005A

The Contractor shall provide hands-on training to designated Government personnel as specified in Section J (Attachment J-9).

C.5.5.2 Develop CGMEA Software User and Operators Procedures, CLIN 0005B

The Contractor shall develop procedures, operating guides, and job-aids for end-users, computer center software operators, preparers and users of application inputs/outputs, and operators of computer hardware. These procedures shall be documented in the *Software User/Operators Manual (SUOM)* and shall be reviewed for acceptance at a *Software Usability Review*.

C.5.6 Data for the CG Mission Essential Application (CGMEA) Training, CLIN 0006

Technical data to be furnished hereunder shall be prepared in accordance with Data Item Descriptions (DIDs) as specified in the CG-SDDS and Section J (Attachment J-7), Contract Data Requirements List (CDRL).

C.5.6.1 CGMEA User/Operator Manual (SUOM), CLIN 0006A [A003]

This Data Item describes format, contents and preparation instructions in support of providing CGMEA applications specific instruction to hands-on users and computer center software operators.

C.5.7 Contract Management, CLIN 0007

- a. The Contractor shall establish and maintain management operations and procedures necessary to control the total software engineering effort and assure effective cost and schedule control of the conversion effort. The Contractor shall document their process and perform quality assurance, configuration management, operational support activities, and

participate in reviews and audits as necessary to accomplish the work defined in this SOW.

- b. The Contractor shall designate a Program Manager (PM) to perform the duties as specified in Section C.7 Contractor Personnel Requirements.

C.5.7.1 Management Process Documentation; Requirements of Software Planning, CLIN 0007A [A001]

The Contractor shall establish and implement a program management office function to manage all technical performance, cost, schedule, and data delivery requirements of the contract. The contractor shall document this process in the *Plans* section of the *Consolidated Software Development Document (CSDD)*.

C.5.7.2 Software Quality Assurance (QA), CLIN 0007B

The contractor shall implement procedures to comply with the software quality requirements of the *SDDS*. The Contractor's Quality Assurance approach is to be documented in the *Plans* section of the *CSDD* as approved and accepted by the Government. This documentation shall constitute a Software Quality Assurance Plan for the purposes of this contract. The contractor may include any formally documented in-house Quality Assurance Plans and procedures.

C.5.7.3 Software Configuration Management (SCM), CLIN 0007C

The contractor shall implement procedures to comply with the software configuration management requirements of the *SDDS*. The Contractor's SCM approach is to be documented in the *Plans* section of the *CSDD* as approved and accepted by the Government. This documentation shall constitute a Configuration Management Plan for the purposes of this contract. The contractor may include any formally documented in-house Software Configuration Management Plans and procedures.

C.5.7.4 Technical Interchange Meetings, Reviews and Audits, CLIN 0007D

The Contractor shall plan for and participate in Joint Technical and Management Reviews and Audits as defined in the SDDS to the level specified in Section J (Attachment J-10) of this contract. The contractor shall schedule formal technical interchange meetings (TIMs) as appropriate throughout this effort.

C.5.7.4.1 Review/Audit Scheduling.

The Contractor shall include in the project schedule all reviews and audits specified in Section J (Attachment J-10) to this contract. Reviews and audits shall coincide with completion of the appropriate documentation and development phases as detailed in the SDDS and this SOW.

C.5.7.4.2 Documentation of Reviews and Audits.

The contractor shall be responsible for submitting agendas in advance of and documenting and submitting minutes resulting from all scheduled TIMs, reviews and audits.

C.5.7.4.3 Independent Verification and Validation (IV&V).

The Government reserves the right to exercise IV&V via Government team or separate contractor to assure contract compliance. In so doing, the Government stipulates that IV&V activities will be conducted as a function of scheduled reviews and audits contained in this contract. If directed, the Contractor shall assist as necessary the IV&V team/contractor during scheduled reviews and audits.

C.5.7.5 Project Status Reporting, CLIN 0007E

The Contractor shall prepare and deliver monthly project status reports in accordance with Exhibit GG of

the Software Development and Documentation Standards (SDDS).

C.5.7.6 Metrics, CLIN 0007F

The contractor shall implement procedures to comply with the management indicator (metric) reporting requirements of the SDDS. The Contractor's management approach to collection and application of metrics is to be documented in the *Plans* section of the *CSDD* as approved and accepted by the Government. At a minimum, the Contractor shall prepare and include in the monthly Project Status Reports, Software Management Indicators (Metrics) listed in Appendix F of the SDDS. In addition, the Contractor may include any formally documented in-house metrics deemed appropriate.

C.5.7.7 Government Directed Travel, CLIN 0007G

The Contractor shall *****G-ACS recommended wording for Government directed travel goes here*****
It is anticipated that contractor personnel may require travel in conjunction with this effort to government facilities outside the Washington Metropolitan area. However, the Contractor shall make every effort to minimize such travel and the number of personnel involved. Facilities anticipated for visit include _____ located in _____, and the _____ located in _____. Travel is to be conducted in accordance with Section H.x.

C.5.8 Data for Project Management, CLIN 0008

Technical data to be furnished hereunder shall be prepared in accordance with Data Item Descriptions (DIDs) as specified in the CG-SDDS and Section J (Attachment J-7), Contract Data Requirements List (CDRL).

C.5.8.1 Monthly Progress Report, CLIN 0008A [CDRL A006]

This Data Item describes format, contents and preparation instructions for progress reports required in support of project management.

C.5.8.2 Technical Interchange Meeting (TIM) and Review Agenda, CLIN 0008B [CDRL A007]

This Data Item describes format, contents and preparation instructions for data generated in support of Technical Interchange Meetings and Reviews.

C.5.8.3 Minutes from TIM, Reviews, and Audits, CLIN 0008C [CDRL A008]

This Data Item describes format, contents and preparation instructions for data generated in support of Technical Interchange Meetings and Reviews.

C.6 APPLICABLE REFERENCE DOCUMENTS

A full list of referenced documents and provisions as to how to obtain those documents are provided in Section J (Attachment J-1) to this contract.

C.7 CONTRACTOR PERSONNEL REQUIREMENTS

- a. The Contractor shall provide and maintain personnel qualified to perform the tasks necessary to deliver the items and services required by this Work Statement.
- b. The Contractor shall supply personnel who are citizens of the United States, with the only exception being legal residents of the United States who may perform duties with ADP-I (non-critical, non-sensitive) or ADP-O (non-ADP) requirements. COMDTINST M5520.12, Coast Guard Personnel Security Program, and DOT Order 1630.5 define the specific criteria for assigning positions (Government and Contractor) to ADP-IV through ADP-I sensitive categories.
- c. The Contractor shall ensure that a stable, competent work force is selected to perform the tasks associated with this Work Statement. All Contractor personnel are required to read, write, communicate and understand English fluently unless waived in writing on an individual

basis by the Contracting Officer. This Work Statement requires performance by persons, who, over the Contract life, increasingly know the USCG organization, missions, IRM architecture, and policies, and its software and hardware environments.

C.7.1 Key Personnel

- a. Key personnel for this Contract are the Contractor Program Manager, Senior Computer Systems Analyst, and Quality Assurance Manager.
- b. Key personnel shall be substituted in accordance with **Section H** of this contract.
- c. Contractor personnel designated as key personnel per **Section H** of this contract will primarily work at the Contractor's facility where the actual development is being performed unless the Contracting Officer grants an individual-based waiver of this requirement.

C.7.2 Personnel Qualifications

- a. Qualifications for Contractor personnel are further detailed below. While separately specified, in all instances, general, specialized, and specific experience may have been gained concurrently.

C.7.2.1 Contractor Program Manager (Key Personnel)

- a. Duties. Serves as the Contractor's program manager, and shall be the Contractor's authorized interface with the Government Contracting Officer (KO), the Contracting Officer's Technical Representative (COTR), and Government management personnel. Responsible for formulating and enforcing work standards, assigning contractor schedules, reviewing work discrepancies, supervising contractor personnel and communicating policies, purposes, and goals of the organization to subordinates. Shall be responsible for the overall contract performance and shall not serve in any other capacity under this contract.
- b. Qualifications. A Bachelor's degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline. This position requires a minimum of ten years experience, of

which at least six years must be specialized.

Specialized experience to include complete engineering project development from inception to deployment, demonstrated ability to provide guidance and direction in tasks similar to the tasks provided in this Statement of Work, proven expertise in the management and control of funds and resources, and demonstrated capability in managing multi-task contracts of this type and complexity. At least one of these years shall have been with the proposed software development environment as specified in Section J (Attachment J-6). General experience includes increasing responsibilities in information systems design and management.

A Master of Science degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline will be considered equivalent to one year specialized experience and two years general experience. A Ph.D. (in the fields described in this paragraph) will be considered equivalent to two years specialized experience and three years general experience.

The Program Manager shall possess a SECRET security clearance. (application specific)

**C.7.2.2 Senior computer Systems Analyst
(Key Personnel)**

- a. Duties. Provides technical and administrative direction for personnel performing software development tasks, including the review of work products for correctness, adherence to the design concept and to user standards, and for progress in accordance with schedules. Coordinates with the Program Manager to ensure problem solution and user satisfaction. Makes recommendations, if needed, for milestone status reports, deliverables, and subordinates, and end user representatives. Provides daily supervision and direction to support staff.
- b. Qualifications. A Bachelor's degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline. This position requires a minimum of eight years experience,

of which at least five years must be specialized.

Specialized experience to include analysis and design of complex business applications utilizing database management systems and programming languages specified in Section J (Attachment J-6). At least one of these years shall have been with the proposed software development environment as specified in Section J (Attachment J-6). Knowledge of current storage and retrieval methods, graphical client/server development, and demonstrated ability to formulate specifications for computer programmers to use in coding, testing and debugging of computer programs. General experience includes increasing responsibilities in assignments of a technical nature. Proven ability to work independently or under only general direction on complex application problems involving all phases of system analysis is required.

A Master of Science degree in Computer Science or Information Systems will be considered equivalent to one year specialized experience and two years general experience. A Ph.D. in Computer Science or Information Systems will be considered equivalent to two years specialized experience and three years general experience.

C.7.2.3 Quality Assurance Manager (Key Personnel)

- a. Duties. Establishes and maintains a process for evaluating software products, development processes, and associated documentation. Determines the resources for quality control. Maintains a high degree of quality throughout the software development life-cycle (SDLC). Conducts formal and informal reviews at pre-determined points throughout the SDLC. Responsible for configuration management processes, software metrics, and software testing and integration.
- b. Qualifications. A Bachelor's degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline. This position requires a minimum of six years experience, of which at least three years must be specialized. Specialized experience to include configuration

management, verification and validation, software testing and integration, software metrics and their application to software quality assessment. General experience includes increasing responsibilities in quality assurance and quality control.

A Master of Science degree in Computer Science or Information Systems will be considered equivalent to one year specialized experience and two years general experience.

C.7.2.4 Computer Systems Analyst

- a. Duties. Analyzes and develops computer software possessing a wide range of capabilities, including numerous engineering, business, and records management functions. Develops plans for automated information systems from project inception to conclusion. Analyzes the business problem, and develops system requirements and program specifications, from which programmers prepare detailed information models, function and process models, data flows, programs, and tests. Coordinates closely with programmers to ensure proper implementation of program and system specifications. Develops, in conjunction with functional users, system alternative solutions.

- b. Qualifications. A Bachelor's degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline. This position requires a minimum of six years experience, of which at least four years must be specialized. Specialized experience to include analysis and design of complex business applications utilizing database management systems and programming languages specified in Section J (Attachment J-6). At least one of these years shall have been with the proposed software development and RDBMS environment as specified in Section J (Attachment J-6). Knowledge of current storage and retrieval methods, one year of system analysis experience designing graphical business applications in a client/server environment.

Demonstrated ability to formulate specifications using IDEF1X and IDEF0 modeling techniques for computer

programmers to use in coding, testing and debugging and maintaining computer programs. General experience includes increasing responsibilities in information systems design and management. Must demonstrate the ability to work independently or under only general direction on requirements that are moderately complex to analyze, plan, program and implement.

A Master of Science degree (in the fields described in this paragraph) will be considered equivalent to one year specialized experience and two years general experience. A Ph.D. in Computer Science or Information Systems will be considered equivalent to two years specialized experience and three years general experience.

C.7.2.5 Senior Software Engineer

- a. Duties. Analyzes complex system and software requirements. Designs and/or implements software tools and subsystems to support software reuse and domain analyses. Manages software development and support using formal specifications, information and process models, other accepted design techniques and Computer Aided Software Engineering (CASE) tools. Estimates software development costs and schedules. Reviews existing and developed programs and assists in making refinements, reducing operating time, and improving current techniques. Supervises software configuration management.
- b. Qualifications. A Bachelor's degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline. This position requires a minimum of six years experience, of which at least three years must be specialized. Specialized experience includes: demonstrated experience in development of large-scale database management systems in a graphical client/server computing environment, working with SQL and third and fourth generation programming languages. At least one of these years shall have been with the proposed software development and RDBMS environment as specified in Section J (Attachment J-6). Must demonstrate the

ability to develop complex software to satisfy design objectives, ability to work independently or under only general direction.

A Master of Science degree in Computer Science, Information Systems or other related scientific or technical discipline will be considered equivalent to one year specialized experience and two years general experience.

C.7.2.6 Application Programmer/Software Engineer

- a. Duties. Analyzes functional business applications and design specifications for functional areas such as accounting, finance, and logistics. Develops block diagrams, information and process models, and logic flow charts. Translates detailed design into computer software. Tests, debugs, and refines the computer software to produce the required product. Prepares required documentation, including both program-level and user-level documentation. Enhances software to reduce operating time and resource requirements, and improve efficiency. Provides technical direction to programmers as required to ensure program deadlines are met.

- b. Qualifications. A Bachelor's degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline. This position requires a minimum of five years experience, of which at least three years must be specialized. Specialized experience to include experience as an applications programmer on large-scale database management systems in a graphical client/server computing environment, knowledge and experience in developing applications to the target CGSWIII application environment as specified in Section J (Attachment J-5). At least one of these years shall have been with the proposed software development and RDBMS environment as specified in Section J (Attachment J-6). Must demonstrate the ability to develop complex software to satisfy design objectives, ability to work independently or under only general direction.

A Master of Science degree in Computer Science,

Information Systems or other related scientific or technical discipline will be considered equivalent to one year specialized experience and two years general experience.

C.7.2.7 Database Management System Specialist

- a. Duties. Manages development of complex database projects. Provides highly technical expertise in the use of relational database management systems. Develops and documents logical and physical database structures to support application development to meet validated user requirements. Defines file organizations, indexing methods, security procedures, and database designs that optimize application performance for specific user applications.

- b. Qualifications. A Bachelor's degree in Computer Science, Information Systems, Engineering, Business, or other related scientific or technical discipline. This position requires a minimum of six years experience, of which at least three years must be specialized. Specialized experience to include demonstrated experience using current RDBMS technologies, application design utilizing various RDBMS and experience with RDBSM internals in graphical, client/server environments. At least one of these years shall have been with the proposed software development and RDBMS environment as specified in Section J (Attachment J-6). General experience includes increasing responsibilities in RDBMS systems analysis and programming. Must demonstrate the ability to develop complex software to satisfy design objectives, ability to work independently or under only general direction.

A Master of Science degree in Computer Science, Information Systems or other related scientific or technical discipline will be considered equivalent to one year specialized experience and two years general experience.

C.7.2.8 Documentation Specialist

- a. Duties. Gathers, analyzes, and composes technical information. Conducts research and ensures the use of proper technical terminology. Translates technical information into clear, readable documents to be used by technical and non-technical personnel. Prepares functional specifications, system specifications, user manuals, special reports and other customer deliverables and documents.
- b. Qualifications. A Bachelor's degree in English, Literature, or other related discipline applicable to this position. This position requires a minimum of three years experience, of which at least one year must be specialized. Specialized experience to include demonstrated experience in preparing technical documentation, technical writing and documentation experience pertaining to all aspects of information systems. Must demonstrate ability to work independently or under only general direction.

C.8 CONTRACTOR SECURITY REQUIREMENTS

C.8.1 General Security

- a. Contract Security Classification Specification (DD Form 254) is applicable and is attached as Attachment J-11.
- b. The Contractor shall comply with the requirements of OMB Circular A-130 in exercising security measures for automated data processing, computer systems and access to the data and facilities.
- c. The Contractor shall provide new systems security and telecommunications network security by following the guidance provided in Chapters 5, 8, 9, 10, and 19 of COMDTINST M5500.13A, and the Privacy Act guidance of COMDTINST M5260.2, Privacy and Freedom of Information Act Manual.
- d. The information systems being built to satisfy the SOW will be unclassified, sensitive information systems. All information or data that they will process, transmit and store will be UNCLASSIFIED.

There will be no national security classified data processed, transmitted or stored. Most of the information will be sensitive data for many reasons, to include FOR OFFICIAL USE ONLY USCG operational, commercial proprietary, investigation sensitive, and Privacy Act. This sensitive data, however, will be used by a few business functions that perform national security-related classified activities. The need to understand the classified aspects of these business functions, in order to properly design these unclassified, sensitive information systems, is what causes the national security personnel and facility requirements of this Contract. Some Task Orders may require access to classified information. Classified information access and security requirements will be furnished on a DD-254, Contract Security Classification Specification, incorporated into those Task Orders requiring access to classified information. Facility and staff security clearances, necessary for performance of a Task Order with a DD-254, shall be in place prior to issuance of a Task Order under this Contract.

- e. The Contractor shall comply with DOD 5220.22-M, Industrial Security Manual for Safeguarding Classified Information, for all matters pertaining to classified information and facility security requirements.
- f. The Contractor shall comply with the Coast Guard Physical Security Manual, COMDTINST M5530.1, for physical security requirements whenever DOD 5220.22-M does not apply.
- g. The Contractor shall comply with the security regulations of the Government installation or Contractor (or subcontractor) facility where on-site work is to be performed.
- h. The Contractor shall not use any electronic information processing equipment in its possession for the purpose of processing or transmitting classified information under this Contract without the written permission of the Contracting Officer.

- i. The Contractor shall not allow employees to use files for logging onto systems that contain the employee's passwords or access controls. The Contractor shall report discovered instances to the COTR and system administrator immediately.
- j. The Contractor shall obtain written approval from the Contracting Officer prior to introducing Contractor owned information system, photo reproduction, telecommunication and data equipment into a Government site. Contractor owned equipment located within a Government site shall be subject to Government security requirements at that site.

C.8.2 Administrative Security - Unclassified Systems

- a. The Contractor shall follow the control principles described in Chapter 7 of COMDTINST M5500.13A, AIS Security Manual. Administrative security consists of management constraints, standard operational security procedures, security accountability controls, and those additional administrative actions necessary to protect IS resources. Administrative security controls define who, what, where, how, and how often actions are taken in the work flow process to ensure the integrity and confidentiality of IS resources. Emphasis shall be placed at those points in the process where control passes from one function, element, or individual to another, as that is where control can be lost.

C.8.3 Malicious Software

- a. The Contractor is specifically prohibited from designing elements into any software which will disable the software in some way, unless required in the functional description. Malicious code set to the conclusion of the warranty or maintenance option shall be precluded for both custom-developed and off-the-shelf portions of the developed software. Delivery by the Contractor to the Government of any software under this Contract and its Task Orders is under an implied certification by the Contractor that the software does not contain malicious code.

C.9 TRANSITION PLANNING AND SUPPORT SERVICES

- a. Software Warranty. All Contractor delivered software associated with this Statement of Work shall be warranted for a period of one year from the date of Government acceptance. All software shall satisfy requirements as specified in the solicitation and the Contractor's accepted Software Plan. Any software deficiencies discovered during the warranty period will be reported to the Contractor in writing. The Contractor shall respond to the deficiencies within 15 working days of notification by the Government. The Contractor shall assess the deficiencies and submit in writing to the Government a proposal for corrective actions. The proposal will be approved by the Government before implementation and all software warranty work by the Contractor shall be at no cost to the Government. **(*****placeholder***may be better in sec H)**
- b. Upon completion of development and successful acceptance testing, the developed CG Mission Essential Application and supporting CSCIs and documentation will transition into the Government's configuration management, operation and maintenance phase. The Contractor shall cooperate fully in any transition of developed software components and associated deliverables to follow-on operations and maintenance activities as directed by the Government.

C.10 NOTES

- C.10.1 The following is a list of attachments contained in Section J to this contract.
 - a. Attachment J-1 contains a full list of referenced documents and provisions as to how to obtain those documents.
 - b. Attachment J-2 contains the definitions of important terms and a listing of acronyms used in this Contract.
 - c. Attachment J-3 specifies Government Furnished Property and Information.

- d. Attachment J-4 contains the Application Performance Specification (including functional requirements) of the CG Mission Essential Application (CGMEA) Software to be converted under this Work Statement.
- e. Attachment J-5 contains the Applications Environment Specification which describes the current source and target environment of the CG Mission Essential Application application.
- f. Attachment J-6 contains the Development Environment Specification which describes software engineering guidelines and constraints, authorized COTS development languages, target RDBMS for development of the CG Mission Essential Application application.
- g. Attachment J-7 lists the Contract Data Requirements List (CDRL) for each data item required of this Statement of Work.
- h. Attachment J-8 specifies Coast Guard sites requiring Government directed, Contractor provided legacy data migration services. **(optional)**
- i. Attachment J-9 contains the specific tasking for Section C.5.5, CLIN 0005 Training.
- j. Attachment J-10 lists the mandatory reviews and audits associated with this Work Statement.
- k. Attachment J-11 contains the Contract Security Classification Specification (DD Form 254).

C.11 PERFORMANCE REQUIREMENTS SUMMARY

- a. Major tasks include the actual conversion of the existing application, legacy data migration, training, and contract management. In addition to the actual accomplishment of the tasks themselves, each task has associated documentation or data deliverables.
- b. The contractor will be testing and integrating into Government provided Operational Test & Evaluation (OT&E) and Run-Time Environments. Details on these environments are provided in Attachment J-5.