

DATA ITEM DESCRIPTION

Title: SOFTWARE DESIGN DESCRIPTION (SDD)

Number: DI-IPSC-81435B

AMSC Number: N10191

DTIC Applicable: No

Preparing Activity: EC

Applicable Forms: None

Approval Date: 20211122

Limitation: None

GIDEP Applicable: No

Project Number: IPSC-2020-002

Use/Relationship

1. The Software Design Description (SDD) describes the design of a Computer Software Configuration Item (CSCI). It describes the CSCI-wide design decisions, the CSCI architectural design, and the detailed design needed to implement the software. The SDD may be supplemented by Interface Design Descriptions (IDDs) (DI-IPSC-81436) and Database Design Descriptions (DBDDs) (DI-IPSC-81437).

1.1. The SDD, with its associated IDDs and DBDDs, is used as the basis for implementing the software. It provides the acquirer visibility into the design and provides information needed for software support.

2. This Data Item Description (DID) contains the format, content, and intended use information for the data product resulting from the work task described in the contract statement of work (SOW).

2.1. This DID is used when the developer is tasked to define and record the design of a CSCI.

2.2. Designs pertaining to interfaces may be presented in the SDD or in IDDs. Designs pertaining to databases may be presented in the SDD or in DBDD.

2.3. The SDD shall be in the format as directed in the contract.

Requirements

3. Format.

a. Automated techniques. Use of automated techniques is encouraged. The term "document" in this DID means a collection of data, regardless of its medium.

b. Alternate presentation styles. Diagrams, tables, matrices, and other presentation styles shall be acceptable substitutes for text when data required by this DID can be made more readable using these styles.

c. Title page or identifier. The document shall include a title page containing, as applicable: document number; volume number; version or revision indicator; security markings in accordance with DoD Manual (DoDM) 5200.01 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 the preparing organization; and distribution statement in accordance with DoD Instruction (DoDI) 5230.24. For data in a database or other alternative form, this information shall be included on external and internal labels or by equivalent identification methods. Copies of the aforementioned instructions can be obtained at <https://www.esd.whs.mil/DD/DoD-Issuances/>

d. Table of contents. 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 in a database or other 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 and labeling. Each page shall contain a unique page number and display the document number, including version, volume, and date, as applicable. For data in a database or other alternative form, files, screens, or other entities shall be assigned names or numbers in such a way that the desired data can be indexed and accessed.

f. Response to tailoring instructions. If a paragraph is tailored out of this DID, the resulting document shall contain the corresponding paragraph number and title, followed by "This paragraph has been tailored out." For data in a database or other 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 the SDD shall be written as multiple paragraphs or subparagraphs if it enhances readability.

h. Standard data descriptions. If a data description required by this DID has been published in a standard data element dictionary specified in the contract, reference to an entry in that dictionary is preferred over including the description itself.

i. Substitution of existing documents. Commercial or other existing documents shall be substituted for all or part of the document if they contain the required data.

Content Requirements

1. Scope. This section shall be divided into the following paragraphs.

1.1. System Identification. This paragraph shall contain a full identification of the

system and the software to which the SDD applies, including, as applicable:

- a) Identification number(s)
- b) Title(s)
- c) Abbreviation(s)
- d) Version number(s)
- e) Release number(s)

1.2. System overview. This paragraph shall briefly state the purpose of the system and the software to which the SDD 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, acquirer, 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 and shall describe any security or privacy considerations associated with its use.

1.4. Section 508, IT Accessibility. Usability will be in compliance with the Section 508 Amendment of the Rehabilitation Act of 1973.

2. Referenced documents. This section shall list the number, title, revision, and date of all documents referenced in this document. This section shall also identify the source for all referenced documents not available through Government sources.

3. CSCI-wide design decisions. This section shall be divided into paragraphs as needed to present CSCI-wide design decisions, that is, decisions about the CSCI's behavioral design (how it will behave, from a user's point of view, in meeting its requirements, ignoring internal implementation) and other decisions affecting the selection and design of the software units that make up the CSCI. If all such decisions are explicit in the CSCI requirements or are deferred to the design of the CSCI's software units, this section shall so state. Design decisions that respond to requirements designated critical, such as those for safety, security, or privacy, shall be placed in separate subparagraphs. If a design decision depends upon system states, modes or lists, explain the software states and modes. Design conventions needed to understand the design shall be presented or referenced. Examples of CSCI-wide design decisions are the following:

a. Input /Output (I/O). Design decisions regarding inputs the CSCI will accept and outputs it will produce, including interfaces with other systems, HWCIs, CSCIs, and users. If part or all of this information is given in IDD, they may be referenced. Define the target operating system(s). CSCI Behavior for I/O Conditions. Design decisions on software behavior in response to each input or condition, including actions the CSCI performs, response times, and other performance characteristics.

b. Equations, algorithms, rules, and handling of un-allowed inputs or conditions.

c. **Databases and Data Files.** Design decisions on how databases and data files appear to the end user. If part or all of this information is given in DBDDs, they may be referenced.

d. **Selected approach to meeting safety, security, and privacy requirements.**

e. **Other Decisions.** Other CSCI-wide design decisions made in response to requirements, such as selected approach to providing required flexibility, availability, and maintainability.

f. **Model Based Systems Engineering (MBSE).** MBSE can be used to develop a set of related system models that help define, design, analyze, and document the system under development.

g. **External Dependencies.** List all external dependencies for example:

1. Software libraries
2. Web services, etc.

3.1. **Cyber security and Personally Identifiable Information (PII) requirements.** This paragraph shall specify the software requirements, concerned with maintaining cyber security and any PII data. These requirements shall include, as applicable, the security/privacy environment in which the software must operate, the type and degree of security or privacy to be provided, the security/privacy risks the software must withstand, required safeguards to reduce those risks, the security/privacy policy that must be met, the security/privacy accountability the software must provide, and the criteria that must be met for security/privacy certification/accreditation.

4. **CSCI architectural design.** This section shall be divided into the following paragraphs to describe the CSCI architectural design. If part or all of the design depends upon system states or modes, this dependency shall be indicated. If design information falls into more than one paragraph, it may be presented once and referenced from the other paragraphs. Design conventions needed to understand the design shall be presented or referenced.

4.1. **CSCI components.** This paragraph shall:

a. Identify the individual software units that make up the CSCI. Each software unit shall be assigned a unique identifier.

Note: A software unit shall be defined as an element in the design of a CSCI; for example, a major subdivision of a CSCI, a component of that subdivision, a class, object, module, function, routine, or database. Software units may occur at different levels of a hierarchy and may consist of other software units. Software units in the design may or may not have a one-to-one relationship with the code and data entities (routines, procedures, databases, datafiles, etc.) that implement them or with the computer files containing those entities. A database may be treated as a CSCI or as a software unit. The SDD may refer to software units by any name(s) consistent with the design methodology being used.

b. Relationships. Show the static (such as "consists of") relationship(s) of the software units. Multiple relationships shall be presented, depending on the selected software design methodology (for example, in an object-oriented design or modular open systems design, this paragraph may present the class and object structures as well as the module and process architectures of the CSCI).

c. Requirements and Allocation. State the purpose of each software unit, the CSCI requirements, and the CSCI-wide design decisions allocated to it.

d. Identifying Information. Identify groups of software unit's development status/type (such as: new development, existing design or software to be reused as is, existing design or software to be reengineered, software to be developed for reuse, software planned for Build N, etc.). For existing design or software, the description shall include identifying information, such as name, version, documentation references, library, etc.

e. Computer Hardware Resources. Describe the CSCI's (and as applicable, each software unit's) planned utilization of computer hardware resources (such as processor capacity, memory capacity, input/output device capacity, auxiliary storage capacity, and communications/network equipment capacity). The description shall cover all computer hardware resources included in resource utilization requirements for the CSCI, in system-level resource allocations affecting the CSCI, and in resource utilization measurement planning in the Software Development Plan. If all utilization data for a given computer hardware resource are presented in a single location, such as in another SDD, the referenced SDD and paragraph shall be stated. Included for each computer hardware resource shall be:

1. The CSCI requirements or system-level resource allocations being satisfied
2. The assumptions and conditions on which the utilization data are based (for example, typical usage, worst-case usage, assumptions of certain events)
3. Any special considerations affecting the utilization (such as use of virtual memory, overlays, or multiprocessors or the impacts of operating system overhead, library software, or other implementation overhead)
4. The units of measure used (such as percentage of processor capacity, cycles per second, bytes of memory, bytes per second)
5. The level(s) at which the estimates or measures are made (such as software unit, CSCI, or executable program)

f. State the library, website, etc. where the software unit is to be placed.

4.2. Concept of execution. This paragraph shall describe the concept of execution among the software units. It shall include diagrams and descriptions showing the dynamic relationship of the software units, that is, how they will interact during CSCI operation, including, as applicable:

- Flow of execution control
- Data flow
- State transition diagrams
- Timing

- Priorities among units
- Handling of interrupts
- Timing/sequencing relationships
- Exception handling
- Concurrent execution
- Dynamic allocation/deallocation
- Dynamic creation or deletion of objects, processes, tasks, and other aspects of dynamic behavior

4.3. Interface design. This paragraph shall be divided into the following subparagraphs to describe the interface characteristics of the software units. It shall include both interfaces among the software units and their interfaces with external entities such as systems, configuration items, Service level Agreements for interfaces, and users. If part or all of this information is contained in IDD's or elsewhere, these sources may be referenced.

4.3.1 Interface identification and diagrams. This paragraph shall state the project-unique identifier assigned to each interface and shall identify the interfacing entities (software units, systems, configuration items, users, etc.) by name, number, version, and documentation references, as applicable. The identification shall state which entities have fixed interface characteristics (and therefore impose interface requirements on interfacing entities) and which are being developed or modified (thus having interface requirements imposed on them). If needed one or more interface diagrams shall be provided, as appropriate, to depict the interfaces.

4.3.2 (Unique identifier of interface). This paragraph shall identify an interface by unique identifier, shall briefly identify the interfacing entities, and shall be divided into subparagraphs, as needed, to describe the interface characteristics of one or both of the interfacing entities. If a given interfacing entity is not covered by this SDD (for example, an external system) but its interface characteristics need to be mentioned to describe interfacing entities that are, these characteristics shall be stated as assumptions or as "When [the entity not covered] does this, [the entity that is covered] will..." This paragraph may reference other documents (such as data dictionaries, standards for protocols, and standards for user interfaces) in place of stating the information here. The design description shall include the following, as applicable, presented in any order suited to the information to be provided, and shall note any differences in these characteristics from the point of view of the interfacing entities (such as different expectations about the size, frequency, or other characteristics of data elements):

- a. Priority assigned to the interface by the interfacing entity(ies) such as:
 1. Spanning tree
 2. Priority queue
 3. Tree
 4. Other data structures
- b. Cybersecurity considerations for interface requirements such as:
 1. Identifying vulnerabilities
 2. Evaluating vulnerabilities

3. Reporting vulnerabilities
 4. Mitigating or resolving security vulnerabilities
- c. Type of interface (such as real-time data transfer, storage-and-retrieval of data, etc.) to be implemented
- d. Characteristics of individual data elements that the interfacing entity(ies) provide, store, send, access, receive, etc., such as:
1. Names and identifiers:
 - a) Unique identifier
 - b) Non-technical (natural-language) name
 - c) DoD standard data element name
 - d) Technical name (e.g., variable or field name in code or database)
 - e) Abbreviation or synonymous name(s)
 2. Data type (alphanumeric, integer, etc.)
 3. Size and format (such as length and punctuation of a character string)
 4. Units of measurement (such as meters, dollars, nanoseconds)
 5. Range or enumeration of possible values (such as 0-99)
 6. Accuracy (how correct) and precision (number of significant digits)
 7. Priority, timing, frequency, volume, sequencing, and other constraints, such as whether the data element may be updated and whether business rules apply
 8. Security and privacy constraints
 9. Sources (setting/sending entities) and recipients (using/receiving entities)
- e. Characteristics of data element assemblies (records, messages, files, arrays, displays, reports, etc.) that the interfacing entity(ies) will provide, store, send, access, receive, etc., such as:
1. Names and identifiers:
 - a) Unique identifier
 - b) Non-technical (natural language) name
 - c) Technical name (e.g., record or data structure name in code or database)
 - d) Abbreviations or synonymous name(s)
 2. Data elements in the assembly and their structure (number, order, grouping)
 3. Medium (such as disk) and structure of data elements/assemblies on the medium
 4. Visual and auditory characteristics of displays and other outputs (such as colors, layouts, fonts, icons and other display elements, beeps, lights)
 5. Relationships among assemblies, such as sorting/access characteristics
 6. Priority, timing, frequency, volume, sequencing, and other constraints, such as whether the assembly may be updated and whether business rules apply
 7. Security and privacy constraints
 8. Sources (setting/sending entities) and recipients (using/receiving entities)
- f. Characteristics of communication methods that the interfacing entity(ies) use for the interface, such as:

1. Unique identifier(s)
2. Communication links, bands, frequencies, media and their characteristics
3. Message formatting
4. Flow control (such as sequence numbering and buffer allocation)
5. Data transfer rate, whether periodic/aperiodic, and interval between transfers
6. Routing, addressing, and naming conventions
7. Transmission services, including priority and grade
8. Safety, security, and privacy considerations, such as encryption, user authentication, compartmentalization, and auditing

g. Characteristics of protocols that the interfacing entity(ies) use for the interface, such as:

1. Unique identifier(s)
2. Priority and layer of the protocol
3. Packeting, fragmentation and reassembly, routing, and addressing
4. Legality checks, error control, and recovery procedures
5. Synchronization, including: connection establishment, maintenance, termination status, identification, and any other reporting features

h. Other characteristics, such as physical compatibility of the interfacing entity(ies) (dimensions, tolerances, loads, voltages, plug compatibility, etc.)

5. Requirements traceability. This section shall contain:

- a. Traceability from each software unit identified in this SDD to the CSCI requirements allocated to it. (Use of MBSE methods is acceptable.)
- b. Traceability from each CSCI requirement to the software units to which it is allocated.

6. Notes. This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, rationale). 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 the SDD.

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.).

End of DI-IPSC-81435B