

NPR 7150.2A Requirements Matrix (Class D and NOT Safety Critical)

Text in black is required by NPR 7150.2A NASA Software Engineering Requirements.

Text in gray is provided only as guidance or suggested content.

Requirement

The Contractor shall flow down all software contractual requirements to all levels of software supplier sub-contracts.

The Contractor shall develop, execute, and maintain the following:

- a. Software Development Plan [defined in Appendix A1]
- b. Software Configuration Management Plan [defined in Appendix A2]
- c. Software Requirements Specification [defined in Appendix A3]
- d. Software Design Description [defined in Appendix A4]
- e. Software Version Description [defined in Appendix A5]
- f. Software Test Plan [defined in Appendix A6]

The Contractor shall ensure that actual milestones and deliverables are tracked against the software plans.

The Contractor shall provide a software schedule for the project to review and specify the frequency at which schedule updates will be provided.

The Contractor shall identify, develop, document, and approve software requirements based on analysis of customer and other stakeholder requirements and the operational concepts.

Guidance: It is highly recommended that an operational concept be documented, reviewed, and approved to ensure that the requirements adequately cover the operational concept.

The Contractor shall ensure that the key units of the software are unit tested.

Guidance: Key units may include: units controlling interfaces, units containing complex algorithms, units with timing constraints or that manage control flow.

The Contractor shall deliver the completed software product(s) and/or data to the customer with appropriate documentation to support the operations and maintenance.

Guidance: The extent of documentation 'to support the operations and maintenance' will depend on the items delivered. For example, a delivered executable may require little or no 'build' documentation. Documentation can take the form of text, audiovisual instructions, interactive scripts, help files, man pages, embedded instructions (e.g. invoked by passing a 'help' argument to the software), etc.

Appendix A1: Software Development Plan

-The Software Development Plan is a living document. Therefore, some of the plan's required content may not be known at the time of its initial release and approval. However, for unknown items, expected closure dates should be assigned and tracked.

a. Project information:

1. Project title
2. Project description
3. Software class = D
4. Safety-criticality determination = Not safety-critical

b. Project Organization.

Project organizational structure showing authority and responsibility of each organizational unit, including contractual suppliers and external organizations (e.g., universities, other government organizations, and industry partners). Include the Software Manager's name and contact information.

Requirement

c. **Engineering environment** Describe software, hardware, and facilities necessary to develop the software products and, as applicable, those items necessary to operate and maintain the software products. This may include computing platform, operating system, libraries (Linear Algebra Package (LAPACK)), equipment (logic analyzer, emulators), standards (Unified Modeling Language Version 2.3, ISO/IEC 14882: Programming Language-C++), procedures (use of software development folders to maintain current development progress), and tools (compiler, debuggers, static analyzers).

d. **Schedule, Effort, and Cost.**

1. Document schedules associated with the activities, services, and deliverables that satisfies the following minimum conditions:

- (a) Coordinates with the overall project schedule if the software is an element of a larger project and
- (b) Documents milestones and delivery dates.

2. Document an estimate of effort (contractor and subcontractor effort).

3. *Document an estimate of special costs. (*if applicable)

Note: Special costs do not include labor and can include (but are not limited to) travel, training, and Off The Shelf (OTS) software or hardware purchases and installation fees. It does not include items provided to but not purchased by the project which may include developer workstations and software development tools.

e. **Requirements Management.** Describe how the project will collect and manage changes to the software requirements. Managing changes to requirements includes determining which changes are feasible, if the changes can be implemented within the approved resources, making all stakeholders aware of the changes, etc.

f. **Make sure the software fulfills its intended use and acceptance criteria.**

4. Document the acceptance criteria that will be used to confirm that the software fulfills its intended use (and any necessary conditions under which the acceptance criteria must be met). The acceptance criteria and conditions may be recorded in this plan, the Software Test Plan and/or in the contractual agreement.

5. Describe how the project will ensure software products meet acceptance criteria and conditions (e.g., by performing acceptance testing, final customer checkout before use, or readiness review).

6. Describe how the project will record results of software acceptance, and address, and track issues to closure (e.g., a test log, defect tracking system, or validation matrix). Describe how the project will perform any additional activities such as demonstration, analysis, or inspection to confirm the product fulfills its intended use and requirements and how the project will record results of those activities, and address, and track issues to closure.

g. **Reviews**

1. Define the reviews that the project will hold of software activities, status, and results with project stakeholders and the frequency or schedule for the reviews.

2. Describe how the project will track issues from reviews to resolution.

Stakeholders are individuals that are affected by or in some way accountable for the outcome of the *project* (may include project members, suppliers, customer/*acquirer*], end users, and others).

Appendix A2: Software Configuration Management Plan

This plan may be included in the Software Management Plan or rolled out as a separate document.

a. **The project name.**

b. **Assigned responsibilities and authority for the implementation of software configuration management on the project.**

c. **All functions and tasks required to manage the configuration of the software, including how the project will:** identify the software configuration items (e.g., software documents, code, data, tools, models, scripts) and their versions to be controlled for the project, and track changes to software products.

d. **Storage location of the software products and identification of any configuration management tools used.**

e. **Plan maintenance information, which identifies the activities and responsibilities necessary to keep the Software Configuration Management Plan up-to-date.**

This could be something as simple as stating: "Changes to the Software Configuration Management Plan will be submitted to and approved by the Software Manager."

f. **Define how the project will document and implement procedures for the storage, delivery, and release of deliverable software products.**

Requirement

Appendix A3: Software Requirements Specification

Guidance: The format and style are left to the project. (* if applicable)

- a. System overview.
- b. Functional requirements and data requirements (e.g., this includes the functions the software must perform and the required inputs, processing, and outputs).
- c. *Required states and modes.
- d. *External interface requirements.
- e. * Adaptation requirements (data used to adapt a program to a given installation site or to given conditions in its operational environment).
- f. * Performance and timing requirements.
- g. * Security and privacy requirements.
- h. * Environment requirements (e.g., computer hardware and operating system).
- i. *Design and implementation constraints (e.g. requirements that constrain the design and construction of the system such as the use of a particular data standard, or programming language, or use of a particular OTS database.)
- j. * Personnel-related requirements (e.g., specific requirements for each user type such as administrator and operator).
- k. * Training-related requirements (e.g., on line help/tutorials).
- l. * Packaging requirements (e.g., storage limitations on deliverable (on one disk), encryption, cyclic redundancy check on the deliverable, or digital signature).
- m. * Testing requirements that drive software design decisions (e.g., checkpoint restart, the ability to see and modify variable values, monitoring execution time or bandwidth, the ability to inject defects).

Appendix A4: Software Design Description

- a. The decomposition into units,
- b. The interrelationship between units,
- c. Concept of execution (e.g., a description or diagram that explains how the units will interact during operation),
- d. External interfaces (including I/O description).

Also consider including the software design decisions (e.g., assumptions, limitations, and reliability related items/concerns or constraints).

Appendix A5: Software Version Description (* if applicable)

- a. Software name and the version identifier to which this SVD applies, (e.g., Software X –Version 9.2, Software X-MM/DD/YY, Software X- Release 2,).
- b. * Summary of updates/changes since the previous SVD, any open defects, and workarounds.
This could be a high-level summary of the enhancements and fixes or the project's current list of changes and defects and their status.
- c. Instructions for building the executable software, including, for example, the instructions and data for compiling and linking and the procedures used for software recovery, software regeneration, testing, or modification.
- d.* If the SVD is not co-located with the software product files, include the location of the files. Software product files are the set of computer programs, procedures, and associated documentation and data. Examples include requirements, design, source code, object code, databases, test information, batch files, command files, data files, manuals, and any files needed to install, build, operate, and maintain the software.

Requirement

Appendix A6: Software Test Plan (* if applicable)

a. Describe how the project will record inputs, record and evaluate test results, document the evaluation, address and track defects to closure and describe where results and evaluations are retained (e.g., a test log, defect tracking system, or validation matrix),

b. Tests cases.

For each test case identify the requirement(s) that the test case verifies and describe the actions necessary to verify the software against the requirement(s).

Actions may include establishing initial conditions, entering inputs/issuing commands, and evaluating the output against expected results and criteria. Multiple test cases may be needed to verify a single requirement; and the test case order may need to be specified.

c. Test milestones

d*. Testing environment(s), site(s), personnel, and participating organizations. This is done when the project needs to plan in advance for a particular software test environment (e.g., the environment does not exist and it must be developed, environments needing the scheduling of particular resources, or specific configuration set up in order to run the tests). The testing environment includes software, hardware, and facilities necessary to run the tests, etc.

Document the test resources and schedule for utilization of those resources (e.g. a lab or simulator or facility). Test preparation and configuration of the environment necessary to run the test can be included here.