

| [NODIS Library](#) | [Program Formulation\(7000s\)](#) | [Search](#) |



NASA Procedural Requirements

COMPLIANCE IS MANDATORY

NPR 7150.2

Effective Date: September 27,
2004

Expiration Date: September
27, 2009

[Printable Format \(PDF\)](#)

Request Notification of Change (NASA Only)

Subject: NASA Software Engineering Requirements

Responsible Office: Office of the Chief Engineer

| [TOC](#) | [Preface](#) | [Chapter1](#) | [Chapter2](#) | [Chapter3](#) | [Chapter4](#) | [Chapter5](#) | [Chapter6](#) | [AppendixA](#)
| [AppendixB](#) | [AppendixC](#) | [AppendixD](#) | [ALL](#) |

CHAPTER 4: Supporting Software Life Cycle Requirements

Support processes typically do not happen in one life cycle phase such as requirements, design, implementation, or test. Support processes typically occur throughout the software life cycle. For example, typical configuration management baselines (e.g., requirements, code, product) happen across the life cycle. Support processes are software management and engineering processes that typically support the entire software life cycle (e.g., configuration management).

4.1 Software Configuration Management

Software configuration management establishes and maintains the integrity of the products of a software project throughout the software life cycle. Software configuration management involves identifying the configuration of products that are delivered to the customer and used in development, systematically controlling changes to the configuration, and maintaining the integrity and traceability of the configuration. Use of standard Center or organizational software configuration management processes and procedures is encouraged where applicable.

4.1.1 The project shall develop a Software Configuration Management Plan that describes the functions, responsibilities, and authority for the implementation of software configuration management for the project. [SWE-079]

Note: The plan may be a part of the project configuration management plan with required content of the plan defined in Chapter 5.

4.1.2 The project shall track and evaluate changes to software products. [SWE-080]

Note: The project can use a Software Change Request or software problem tracking system. Software Change Requests/Problem Reports should be documented per the requirements in Chapter 5.

4.1.3 The project shall identify the software configuration items (e.g., software documents, code, data, scripts) and their versions to be controlled for the project. [SWE-081]

4.1.4 The project shall establish and implement procedures designating the levels of control each identified configuration item must pass through; the persons or groups with authority to authorize changes and to make changes at each level; and the steps to be followed to request authorization for changes, process Change Requests, track changes, distribute changes, and maintain past versions. [SWE-082]

4.1.5 The project shall prepare and maintain records of the configuration status of configuration items. [SWE-083]

Note: Configuration status accounting generates and/or maintains records of the status and contents of the software throughout the life cycle. This function keeps track of the changes and the contents of versions and releases.

4.1.6 The project shall ensure that software configuration audits are performed to determine the correct version of the configuration items and verify that they conform to the documents that define them. [SWE-084]

4.1.7 The project shall establish and implement procedures for the storage, handling, delivery, release, and maintenance of deliverable software products. [SWE-085]

4.2 Risk Management

Identification and management of risks provides a basis for systematically examining changing situations over time to uncover and correct circumstances that impact the ability of the project to meet its objectives.

4.2.1 The project shall identify, analyze, plan, track, control, communicate, and document software risks (potential issues, hazards, threats, and vulnerabilities) in accordance with NPR 7120.5, NASA Program and Project Management Processes and Requirements and NPR 8000.4, Risk Management Procedural Requirements. [SWE-086]

4.3 Peer Reviews/Inspections

Peer reviews and inspections are the in-process technical examination of work products by the supplier's peers for the purpose of finding and eliminating defects early in the life cycle. Peer reviews are performed following defined procedures covering the preparation for the review, conducting the review itself, documenting results, reporting the results, and certifying the completion criteria. Peer reviews and inspections which satisfy these features include, but are not limited to, Fagan Inspections, Software Formal Inspections, Tom Gilb's Software Inspections, and Perspective Based Reading.

4.3.1 The project shall ensure peer reviews are performed for: [SWE-087]

- a. Software Requirements.
- b. Software Test Plans.
- c. Any design and code items that the project identified for peer review according to the software development plans.

Note: Safety and mission-success related design and code components should be peer reviewed.

4.3.2 The project shall, for each planned peer review: [SWE-088]

- a. Use a checklist to evaluate the work products.
- b. Use established readiness and completion criteria.
- c. Track actions identified in the reviews until they are resolved.

4.3.3 The project shall, for each planned peer review, record basic measurements. [SWE-089]

Note: The requirements for the content of the Software Inspection/Peer Review Report are defined in Chapter 5.

4.4 Software Measurement

Software measurement programs at multiple levels should be established to meet measurement objectives that are derived from identified information needs and objectives. The requirements below are designed to establish measurement programs at the project and the Mission Directorate levels to assist in managing projects, assuring quality, and improving software engineering practices. Project-level and Mission Directorate-level (product line) measurement programs should be designed to meet the following high-level goals:

- a. To improve future planning and cost estimation.
- b. To provide realistic data for progress tracking.
- c. To provide indicators of software quality.
- d. To provide baseline information for future process improvement activities.

Additional measures can be defined by either the projects or the Mission Directorate, based on any additional high-level goals they may have.

4.4.1 The project shall establish and document specific measurement objectives for their project. [SWE-090]

4.4.2 The project shall select and record the selection of specific measures in the following areas: [SWE-091]

- a. Software progress tracking.
- b. Software functionality.
- c. Software quality.

d. Software requirements volatility.

e. Software characteristics.

Note: Metrics reports should be documented per the metrics report requirements of Chapter 5.

4.4.3 The project shall specify and record data collection and storage procedures for their selected software measures and collect and store measures accordingly. [SWE-092]

Note: Data should be maintained in the NASA process asset library.

4.4.4 The project shall analyze software measurement data collected using documented project-specified and Center/organizational analysis procedures. [SWE-093]

4.4.5 The project shall report measurement analysis results periodically and allow access to measurement information by Center-defined organizational measurement programs. [SWE-094]

4.4.6 Each NASA Mission Directorate shall establish its own software measurement system to include the minimum reporting requirements in SWE-091. [SWE-095]

4.4.7 Each NASA Mission Directorate shall identify and document the specific measurement objectives, the chosen specific measures, the collection procedures, and storage and analysis procedures. [SWE-096]

4.4.8 Each NASA Mission Directorate shall report their software measurement results to the Office of the Chief Engineer on a yearly basis. [SWE-097]

4.5 Best Practices

Successful best practices throughout the software community provide an available resource that can lead to improved products. Ensuring an awareness of these practices can often provide potential solutions to problems. Successful best practices also provide alternate approaches for an individual project to consider, given its scope, domain, and goals. The intent of organizational best practices is not to mandate the use of any specific practice, but to provide information and examples to each project so that it can evaluate and choose those practices that it deems most beneficial.

4.5.1 The NASA Office of the Chief Engineer shall maintain an Agencywide process asset library of applicable best practices. [SWE-098]

Note: The repository may contain information in many forms including, but not limited to, websites, design principles, books, periodicals, presentations, and conference descriptions.

4.5.2 Each Center shall review the contents of the process asset library to identify those practices that may have direct applicability and value to its software activities. [SWE-099]

4.6 Training

Properly trained personnel are key to success with software engineering projects. The goal is to maintain and advance organizational capability for training of personnel that perform software engineering practices to effectively meet scientific and technological objectives. The Software Training Plan should include training in the following software activities: software management, software acquisition, software monitoring, software development, software safety and mission assurance, and software process improvement.

4.6.1 The NASA Chief Engineer and Center training organizations shall provide and fund training to advance software engineering practices and software acquisition. [SWE-100]

4.6.2 Each Center shall maintain and implement a Software Training Plan(s) to advance its in-house software engineering capability and as a reference for its contractors. [SWE-101]

Note: The Software Training Plan should be documented per the Software Training Plan requirements of Chapter 5. Centers should plan to meet or exceed the CMMI® SE/SW Maturity Level 3.

| [TOC](#) | [Preface](#) | [Chapter1](#) | [Chapter2](#) | [Chapter3](#) | [Chapter4](#) | [Chapter5](#) | [Chapter6](#) |
| [AppendixA](#) | [AppendixB](#) | [AppendixC](#) | [AppendixD](#) | [ALL](#) |

| [NODIS Library](#) | [Program Formulation\(7000s\)](#) | [Search](#) |

DISTRIBUTION:
NODIS

This Document Is Uncontrolled When Printed.
Check the NASA Online Directives Information System (NODIS) Library
to Verify that this is the correct version before use: <http://nodis3.gsfc.nasa.gov>
