



# NASA Procedural Requirements

**COMPLIANCE IS MANDATORY FOR NASA EMPLOYEES**

**NPR 7150.2D**

Effective Date: March 08,  
2022

Expiration Date: March 08,  
2027

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

---

## Subject: NASA Software Engineering Requirements

Responsible Office: Office of the Chief Engineer

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

## Chapter 5: Supporting Software Life Cycle Requirements

### 5.1 Software Configuration Management (SCM)

5.1.1 Software Configuration Management (SCM) is the process of applying configuration management throughout the software life cycle to ensure the completeness and correctness of software configuration items. SCM applies technical and administrative direction and surveillance to identify and record the functional and physical characteristics of software configuration items, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements. SCM establishes and maintains the integrity of the products of a software project throughout the software life cycle. Use of standard Center or organizational SCM processes and procedures is encouraged where applicable.

5.1.2 The project manager 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]

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

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

*Note: The items to be controlled include tools, items, or settings used to develop the software, which could impact the software. Examples of such items include compiler/assembler versions, makefiles, batch files, and specific environment settings.*

5.1.5 The project manager shall establish and implement procedures to: [SWE-082]

- a. Designate the levels of control through which each identified software configuration item is required to pass.
- b. Identify the persons or groups with authority to authorize changes.
- c. Identify the persons or groups to make changes at each level.

*Note: IEEE 828-2012, IEEE Standard for Configuration Management in Systems and Software Engineering describes configuration management processes to be established, how they are to be accomplished, who is responsible for doing specific activities, when they are to happen, and what specific resources are required. It addresses configuration management activities over a product's life cycle. Configuration management in systems and software engineering is a specialty discipline within the larger discipline of configuration management. Configuration management is essential to systems engineering*

and software engineering.

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

5.1.7 The project manager shall perform software configuration audits to determine the correct version of the software configuration items and verify that they conform to the records that define them. [SWE-084]

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

5.1.9 The project manager shall participate in any joint NASA/developer audits. [SWE-045]

## 5.2 Software Risk Management

The project manager shall record, analyze, plan, track, control, and communicate all of the software risks and mitigation plans. [SWE-086]

*Note: Project managers should be aware of any risks that remain after mitigations have been completed or after a risk has been accepted.*

## 5.3 Software Peer Reviews and Inspections

5.3.1 Software peer reviews and inspections are the in-process technical examination of work products by peers to find and eliminate defects early in the life cycle. Software peer reviews and inspections are performed following defined procedures covering the preparation for the review, the review itself is conducted, results are recorded, results are reported, and completion criteria is certified. When planning the composition of a software peer review or inspection team, consider including software testing, system testing, software assurance, software safety, software cybersecurity, and software IV&V personnel.

5.3.2 The project manager shall perform and report the results of software peer reviews or software inspections for: [SWE-087]

- a. Software requirements.
- b. Software plans, including cybersecurity.
- c. Any design items that the project identified for software peer review or software inspections according to the software development plans.
- d. Software code as defined in the software and or project plans.
- e. Software test procedures.

*Note: Software peer reviews or software inspections are recommended best practices for all safety and mission-success related software components. Recommended best practices and guidelines for software formal inspections are contained in NASA-STD-8739.9, Software Formal Inspection Standard.*

5.3.3 The project manager shall, for each planned software peer review or software inspection: [SWE-088]

- a. Use a checklist or formal reading technique (e.g., perspective-based reading) to evaluate the work products.
- b. Use established readiness and completion criteria.
- c. Track actions identified in the reviews until they are resolved.
- d. Identify the required participants.

5.3.4 The project manager shall, for each planned software peer review or software inspection, record necessary measurements. [SWE-089]

## 5.4 Software Measurements

5.4.1 Software measurement is a primary tool for managing software processes and evaluating the quality of software products. Analysis of measures provides insight into the capability of the software organization and identifies opportunities for software process and product improvements. Software measurement programs at multiple levels are established to meet measurement objectives. The requirements below are designed to reinforce the use of measurement at the project, Center software organization, and NASA Chief Engineer levels to assist in managing projects, assuring quality, and improving software engineering practices. Measurement programs are designed to meet the following goals:

- a. Improve future software planning and software cost estimation.
- b. Describe and record information about a software product during its life cycle.
- c. Assist usability and maintainability of a software product.
- d. Monitor and control software life cycle processes.
- e. Communicate information about the system, software product, or service.
- f. Provide a history, including lessons learned, during the development and maintenance to support management and process improvement.
- g. Provide evidence that the processes were followed.
- h. Provide indicators of software quality.
- i. Track the status of software engineering improvement and assurance programs.
- j. Report the status of software engineering improvements and assurance programs to Center software organizations and Center SMA.

5.4.2 The project manager shall establish, record, maintain, report, and utilize software management and technical measurements. [SWE-090]

*Note: The NASA-HDBK-2203 contains a set of candidate management indicators that may be used on a software development project. The NASA Chief Engineer may identify and document additional Center measurement objectives, software measurements, collection procedures and guidelines, and analysis procedures for selected software projects and software development organizations. The software measurement process includes collecting software technical measurement data from the project's software developer(s).*

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

5.4.4 The project manager shall provide access to the software measurement data, measurement analyses, and software development status as requested to the sponsoring Mission Directorate, the NASA Chief Engineer, the Center Technical Authorities, HQ SMA, and other organizations as appropriate. [SWE-094]

5.4.5 The project manager shall monitor measures to ensure the software will meet or exceed performance and functionality requirements, including satisfying constraints. [SWE-199]

*Note: The metrics could include planned and actual use of computer hardware resources (such as processor capacity, memory capacity, input/output device capacity, auxiliary storage device capacity, and communications/network equipment capacity, bus traffic, partition allocation) over time. As part of the verification of the software detailed design, the developer will update the estimation of the technical resource metrics. As part of the verification of the coding, testing, and validation, the technical resource metrics will be updated with the measured values and will be compared to the margins.*

5.4.6 The project manager shall collect, track, and report software requirements volatility metrics. [SWE-200]

## 5.5 Software Non-conformance or Defect Management

5.5.1 The project manager shall track and maintain software non-conformances (including defects in tools and appropriate ground software). [SWE-201]

5.5.2 The project manager shall define and implement clear software severity levels for all software non-conformances (including tools, COTS, GOTS, MOTs, OSS, reused software components, and applicable ground systems). [SWE-202]

*Note: At a minimum, classes should include loss of life or loss of vehicle, mission success, visible to the user with operational workarounds, and an 'other' class that does not meet previous criteria.*

5.5.3 The project manager shall implement mandatory assessments of reported non-conformances for all COTS, GOTS, MOTs, OSS, and/or reused software components. [SWE-203]

*Note: This includes operating systems, run-time systems, device drivers, code generators, compilers, math libraries, and build and Configuration Management (CM) tools. It should be performed pre-flight, with mandatory code audits for critical defects.*

5.5.4 The project manager shall implement process assessments for all high-severity software non-conformances (closed loop process). [SWE-204]

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

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

**DISTRIBUTION:**  
**NODIS**

---

**This document does not bind the public, except as authorized by law or as incorporated into a contract. 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: <https://nodis3.gsfc.nasa.gov>.**

---