ARMSTRONG FLIGHT PROCEDURAL REQUIREMENTS (AFPR)	**Directive:**	**AFPR-7150.2-001, Revision C**
	Effective Date:	**July 30, 2025**
	Expiration Date:	**July 30, 2030**

Compliance is mandatory.

SUBJECT: Armstrong Software Engineering (SWE) Requirements

RESPONSIBLE OFFICE: Office of the Chief Engineer

Table of Contents

PREFACE

P.1 Purpose

a. This Armstrong Flight Research Center (AFRC) (hereinafter referred to as Center) in Edwards, California, Procedural Requirement (AFPR) brings the Center into compliance with the National Aeronautics and Space Administration (NASA) Policy Directive 7120.4, NASA Engineering and Program/Project Management Policy, by capturing the requirements in NASA Procedural Requirements (NPR) 7150.2, NASA Software Engineering Requirements, and NASA-Standard (STD)-8739.8, NASA Software Assurance and Software Safety Standard. In doing so, this directive provides requirements for the specification, acquisition, development, maintenance, operation, and management of software that supports the Center's flight research mission. It does not prescribe or promote a specific software development life cycle, but instead provides a single set of requirements for Center software engineering (SWE) activities. This will allow organizations at the Center that purchase or develop software the freedom to develop processes tailored for their mission need.

b. In addition to the above, this directive modifies the software classification approach defined in NPR 7150.2 to a hazard- / risk-based system. The approach used in this directive is consistent with the software classification approach defined in NPR 7150.2 for aeronautics applications. This has been done to reduce confusion and improve traceability to other common aeronautics standards and existing Center processes, including Radio Technical Commission for Aeronautics (RTCA) DO-178, Software Considerations in Airborne Systems and Equipment Certification, and Armstrong Flight Research Center Guide (AFG)-7900.3-031, Hazard Management Guide.

P.2 Applicability

a. This directive is applicable to the Center and on-site support contractors, grant recipients, and other partners, developed or revised after the effective date of this AFPR, to the extent specified in their contracts or agreements.

> *Note: The above statement alone is not sufficient to stipulate requirements for the contractor, grant recipient, or agreement. This directive provides requirements for NASA contracts, grant recipients, or agreements to the responsible NASA Project Managers (PM) and contracting officers that are made mandatory through contract clauses, specifications, or statements of work (SOW) in conformance with the NASA Federal Acquisition Regulation (FAR) Supplement (NFS) or by stipulating in the contracts, grants, or agreements which of the NPR 7150.2 requirements apply.*

b. This language applies to Jet Propulsion Laboratory (a Federally-Funded Research and Development Center), other contractors, recipients of grants, cooperative agreements, or other agreements only to the extent specified or referenced in the applicable contracts, grants, or agreements.

c. This directive applies to the development, acquisition, and management of flight, flight support, and ground software. This includes flight software, simulation software, range software, and analysis tools. These can be developed by the Center, Inter-centers, maintained by the Center, acquired by the Center, implemented on Center assets (including flight vehicles, simulations, and range assets) or maintained for the Center by another organization (either at the Center or at a remote site).

d. This directive applies to software development, maintenance, retirement, operations, management, acquisition, and assurance activities. The requirements of this directive cover all software created, acquired, or maintained by or for NASA and apply to all of the Agency's investment areas containing software systems and subsystems. The applicability of these requirements to specific systems and subsystems within the Center's investment areas, programs, and projects is determined by using the NASA-wide definition of software classes and safety criticality, as detailed in this directive, in conjunction with Appendix D, Requirements Mapping Matrix, of this directive. Some projects may contain multiple systems and subsystems having different software classes. Using Appendix D of this directive, the applicable requirements and their associated rigor are adapted according to the classification and safety criticality of the software.

e. This directive will be applied to software development, maintenance, operations, management, acquisition, and assurance activities started after its effective date of issuance.

f. This directive does not supersede more stringent requirements imposed by individual NASA organizations and other Federal government agencies.

g. In this directive, all mandatory actions (i.e., requirements) are denoted by statements containing the term "shall." The terms "may" or "can" denote discretionary privilege or permission, "should" denotes a good practice and is recommended, but not required, "will" denotes expected outcome, and "are / is" denotes descriptive material.

h. In this directive, refer to Appendix A for the contextual understanding of terms used, such as "software," "software engineering," "Software Independent Technical Authority (ITA)," and "Software Technical Authority (TA)."

i. In this directive, all document citations are assumed to be the latest version unless otherwise noted.

j. In this directive, the term "simulation" refers to only those simulations that are implemented in software.

P.3 Authority

a. NPR 7150.2, NASA Software Engineering Requirements

b. NPR 2810.1, Security of Information and Information Systems

c. NASA-STD-8739.8, Software Assurance and Software Safety Standard

P.4 Applicable Documents and Forms

a. NPR 1441.1, NASA Records Management Program Requirements

b. NPR 2210.1, Release of NASA Software

c. NPR 2800.1, Managing Information Technology

d. NPR 7120.7, NASA Information Technology and Institutional Program and Project Management Requirements

e. NPR 7120.8, NASA Research and Technology Program and Project Management Requirements

f. NPR 7123.1, NASA Systems Engineering Processes and Requirements

g. NPR 7900.3, Aircraft Operations Management

h. NPR 8621.1, NASA Procedural Requirements for Mishap and Close Call Reporting, Investigating, and Recordkeeping

i. NPR 8715.3, Requesting Relief from Agency Mission Assurance Requirements

j. Armstrong Flight Research Center Policy Directive (AFPD)-1000.0-002, Governance and Strategic Management

k. AFPD-8040.1-001, Configuration Management

l. AFPD-8700.1-001, Organizational & Individual Safety Responsibilities

m. AFPR-7123.1-001, Systems Engineering Requirements Document

n. AFPR-7123.2-001, Waivers and Deviations to Technical Requirements and Standards

o. AFRC 8621.1-001, Armstrong Flight Research Center – Mishap Preparedness and Contingency Plan

p. AFOP-7120.5-003, Program and Project Management Manual

q. AFOP 7150.2-004, Software Assurance

r. AFOP-7900.3-023, Airworthiness and Flight Safety Review Process

s. AFOP-7900.3-024, Flight Operational Readiness Review (ORR)

t. AFG-7900.3-031, Hazard Management Guide

u. AFG-8739.8-002, Software Assurance Audit and Corrective Action Handbook

v. NASA-STD-7009, Standard for Models and Simulations

w. NASA-STD-8739.9, Software Formal Inspections Standard

x. NASA-Handbook (HDBK)-2203, NASA Software Engineering Handbook (SWEHB)

y. NASA-HDBK-4008, Programmable Logic Devices (PLD) Handbook

z. NASA-HDBK-8739.23, NASA Complex Electronics Handbook For Assurance Professionals

aa. NASA/SP-2010-3403, NASA Scheduling Management Handbook

bb. NASA-GB-8719.13, NASA Software Safety Guidebook

cc. AFRC 10117f, Request for Deviation or Waiver

dd. AFRC 70010, Configuration Change Request

ee. AFRC 80184, Flight Media Release

ff. Radio Technical Commission for Aeronautics (RTCA) DO-178 Rev C, Software Considerations in Airborne Systems and Equipment Certification

gg. Requirements and Standards International Standards Organization (ISO) / International Electrotechnical Commission (IEC) Electronics / Institute of Electrical and Electronics Engineers (IEEE) 2382-20, Information Technology – Vocabulary – Part 20: System Development

hh. ISO/IEC/IEEE 24765:2010 System and Software Engineering – Vocabulary

ii. FAR 2.101

jj. IEEE 1012, Standard for Software Verification and Validation

kk. IEEE 1028, Standard for Software Reviews and Audits

P.5 Measurement / Verification

The methods to ensure compliance with this directive and NPR 7150.2 will be documented in the software development implementation procedures and through internal and external assessments and audits.

P.6 Cancellation

AFPR-7150.2-001 B-7, Armstrong Software Engineering (SWE) Requirements, dated July 23, 2025

/s/ Center Director

DISTRIBUTION: Approved for release via the Document Library.

Chapter 1: Introduction

1.1 Sources of Requirements

This directive seeks to provide a unified set of process requirements for software development / management activities at the Center. It includes the tailored SWE requirements specified in NPR 7150.2, and the software safety requirements specified in NASA-STD-8739.8, Software Assurance and Software Safety Standard. The software guide now resides in NASA-HDBK-2203, NASA Software Engineering and Assurance Handbook (SWEHB). This directive also includes requirements derived from RTCA DO-178, as well as Center-unique requirements needed to ensure airworthiness.

1.2 Document Scope

This directive represents the Center's tailored approach to interpreting the NASA requirements specified in NPR 7150.2 and NASA-STD-8739.8. The scope of this directive only includes NPR 7150.2 Classes C, D, and E software requirements. The scope does not include NPR 7150.2 Class A, B, and F software requirements since the Center does not currently generate this level of software. For software that is classified into NPR 7150.2 Class A, B, and F please refer directly to NPR 7150.2.

This document does not reference or include software components developed with the assistance of artificial intelligence (AI) tools. However, if AI tools are used in the development, modification, or generation of any software elements—such as code, logic, or documentation—the resulting content is subject to the same NPR 7150.2 software requirements. In such cases, users will coordinate with their designated Software ITA and Software Quality Assurance (SQA) to ensure compliance with applicable software engineering standards and requirements, cybersecurity requirements, regulatory obligations, and system integrity expectations.

1.3 Description of Scope

For the purposes of this directive, the definition of "software" is derived from NPR 7150.2 (see Appendix A of this directive) and includes software executing on processors embedded in programmable logic devices.

Based on this definition, types of software include, but are not limited to:

a. Application software: Software designed to help users perform particular tasks or handle particular types of problems, as distinct from software that controls the computer itself.

b. Custom software: Software product developed for a specific application from a user-requirements specification.

c. Configuration software: Software used to define, modify, or manage the operational settings, parameters, or options of a system or component without altering its underlying source code. Configuration software enables system customization, integration, and behavior control within defined limits.

d. Embedded software: Software that is part of a larger system and performs some of the requirements of that system.

e. Existing software: Software that is already developed and available, is usable either as-is or with modifications, and that is provided by the supplier, acquirer, or a third party.

f. Firmware: Combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device.

g. Programmable Logic Device: Field Programmable Gate Array and Complex Programmable Logic Device (see NASA-HDBK-4008).

h. Software reuse: A software product developed for one use, but having other, or one developed specifically to be usable on multiple projects or in multiple roles on one project. Examples include, but are not limited to, Commercial Off-The-Shelf (COTS) products, acquirer-furnished software products, software products in reuse libraries, and pre-existing developer software products. Each use may include all or part of the software product and may involve its modification. This term can be applied to any software, such as requirements and architectures, not just to software code itself. Often, this is software previously written by an in-house development team and used on a different project. Government Off-The-Shelf (GOTS) software would come under this category if the product is supplied from one government project to another government project.

i. Software tool: A computer program used in the development, testing, analysis, or maintenance of a program or its documentation.

j. Support software: Software that aids in the development or maintenance of other software.

k. System software: Software designed to facilitate the operation and maintenance of a computer system and associated programs. (Reference: ISO / IEC / IEEE 24765:2010)

l. Legacy and heritage: Software products (i.e., architecture, code, requirements) written specifically for one project and then, without prior planning during its initial development, found to be useful on other projects.

Software can be compiled or interpreted. Interpreted software includes scripting (i.e., shell scripts, test scripts within a simulation, parameter or preference files, spreadsheets used for data analysis, etc.).

Chapter 2: Responsibilities

The designated governance framework for the Center is defined in AFPD-1000.0-002, Governance and Strategic Management. NPR 7150.2 provides the tailoring, engineering TA, and compliance requirements. Chapter 4 of this directive identifies the TA for each of the NPR 7150.2 requirements. In the case of NPR 7150.2, the Center-level TA is delegated to the Center Director, or the Center Director's designated Engineering TA. Implementation of the NASA software safety standard requirements is the responsibility of the Safety & Mission Assurance (S&MA) Directorate. This Directorate ensures that the requirements found in NASA-STD-8739.8 are being met by the Center.

The program roles are divided amongst the following personnel listed below and their responsibilities explained in the following subsections:

a. Project Manager (PM)

b. Project Chief Engineer (PCE)

c. Operations Manager / Lead

d. System Safety Manager

e. Software Assurance Manager

f. Software Manager

g. Configuration Manager

h. Flight Systems Lead (FSL)

2.1 Project Manager (PM)

The PM is responsible to:

a. Define project priorities, objectives, and milestones in the project plan; and ensure the allocation of resources in the form of aircraft, schedule, staffing, and facilities.

b. Ensure the project plan, request for proposal, contract proposals, and SOW contain appropriate software assurance provisions. The project plan will identify the software manager or their organization and designate a software development agent (SDA). Provisions include:

(1) Safety critical software development / management delegated to the project are produced in accordance with the requirements found in Section 4.4 of this directive with notation of "Safety Critical."

(2) Software deliverables.

(3) Customer surveillance for software assurance in Chapter 5 of this directive and AFOP-7150.2.

(4) How the contractor and customer report and resolve software problems.

(5) Customer agreement for any changes to baselined software elements.

c. Approve all plans generated by the project, including the software development plan (SDP) / software management plan (SMP), software assurance plan (SAP), and configuration management (CM) plan (CMP).

d. Appoint the configuration control board (CCB) membership.

e. Chair, manage, and preside over the CCB.

f. Approve decisions made by the members of the CCB.

g. Integrate the software safety program with system safety and software development.

h. Work with S&MA to acquire software, resolve conflicts, and implement a process for tracking concerns and hazards management.

2.2 Project Chief Engineer (PCE)

The PCE has three main areas of responsibility as technical lead, system safety lead, and systems engineering (SE) lead. On some projects, the PCE may also have some responsibility for doing the work of a technical discipline on the project. On large projects, the PCE may delegate some of the technical lead responsibilities to a deputy or, some of the SE lead responsibilities to a systems engineer or a deputy. Regardless of who does the actual work, the PCE is ultimately responsible.

As technical lead, the PCE is accountable for developing technical objectives, giving technical briefings / presentations, leading experiment, and system design, managing technical risk, supporting test planning and preparation, directing project flight operations, and ensuring appropriate documentation and reporting. The PCE is also responsible for developing the engineering team organization, providing direction for project engineering activities, supporting schedule maintenance, executing the technical work of the project, and working with the Research & Engineering Directorate Branch Chiefs to ensure that the project is adequately staffed.

As system safety lead, the PCE is accountable for the principle technical focus in all areas of hazard analysis efforts and serving as the focal point for safety concerns. The S&MA Directorate typically provides a system safety representative to the project and,

while that person often facilitates System Safety Working Group (SSWG) meetings and prepares the hazard reports and associated materials, the PCE is ultimately accountable and responsible to ensure that hazards are identified, documented, and mitigated appropriately.

As SE lead, the PCE is accountable for ensuring the implementation of applicable SE policies, practices, processes, and procedures throughout the project life cycle, including reviews and required documentation.

2.3 Operations Manager / Lead

The Operations Engineering Branch has primary responsibility for overall surveillance of aircraft configuration and, in conjunction with quality assurance, is responsible for determining that the aircraft software has been properly generated, verified, and validated, and, therefore, acceptable for flight.

An operations engineer's core responsibility is to ensure airworthiness, as defined in Appendix A of this directive, of aircraft and research / science assets by:

a. Approves work orders to load flight and flight support software on board the test vehicle.

b. Develops flight test plans and cards.

c. Ensures operational documents describe all safety-related commands, data, input sequences, and options.

d. Ensures operational documents include error message descriptions and corrective actions.

e. Ensures the test vehicle meets its functional or physical configurations, if applicable.

Authority for issues that relate to the application of aircraft airworthiness standards for modification, operation, or maintenance of aircraft are delegated from the Center Director to the Director of Flight Operations, down to the operations engineer per AFPR-7123.1-001, Systems Engineering Requirements Document. This authority is exercised in different ways depending on the focus area an operations engineer is assigned. Aeromechanical design engineers ensure mechanical designs are designed and fabricated using appropriate standards and static and dynamic stress margins. Drawing control and CM operations engineers ensure drawings and CM documents meet Center requirements for proper tracking, documentation, and archival. As for the conventional operations engineer, the delegation of authority makes them the Flight Operations Directorate representative for project and aircraft work ensuring that work is planned and accomplished based on maintenance, Center, and project requirements as they relate to aircraft integration.

2.4 System Safety Manager

The Flight Research & Test Safety Branch Chief designates a qualified branch member to fulfill the position of System Safety Manager on each project. The System Safety Manager has the following roles and responsibilities:

a. Supports project by attending project meetings and providing an assessment of project compliance with applicable safety requirements to project management and the Flight Research & Test Safety Branch Chief.

b. Develops and maintains System Safety Plan (SSP) and participates with project in development of Risk Management Plan (RMP).

c. Reviews and / or assists with the Preliminary Hazard Analyses (PHA) (iterative process).

d. Participates in design reviews.

e. Reviews and / or assists with development of System Hazard Analysis (SHA) / Sub-SHA / Operating and Support Hazard Analysis (iterative process).

f. Assists with development of Hazard Action Matrix and accepted risks.

g. Participates in CCB meeting as required for system safety / software assurance issues from project conception through end of flight activities.

h. Presents applicable system safety / software assurance risk analysis documentation portions of Flight Readiness Review project briefing and technical briefs.

i. Verifies and documents the status of corrective actions in Flight Assurance Matrix (FAM).

j. Participates in crew briefs and mission control room test and training activities based on general responsibilities identified and agreed upon.

k. Provides continual verbal / written feedback to Flight Research & Test Safety Branch Chief on project safety and risk issues, and compliance with standards.

l. Updates existing hazard reports with any newly identified flight or ground safety risks, prepares materials to brief senior management of any changes to the current accepted level of residual, and updates the FAM accordingly.

m. Assists project in developing, documenting, and processing lessons learned.

2.5 Software Assurance Manager

The Flight Research & Test Safety Branch Chief designates a qualified branch member to fulfill the position of Software Assurance Manager on each project. The Software Assurance Manager has the following roles and responsibilities:

a. Verifies Center directives and NASA standards (or equivalents) in contract / memorandum of agreement / memorandum of understanding, in addition to safety contents.

b. Performs an independent software classification to compare with the software manager's or SDA's classification.

c. Prepares the SAP to establish and implement the software assurance program.

d. Reviews and concurs with the software classification, project plan or SMP, provider SDP, SSP, software requirement document, software design document (SWDD), and software test documentation.

e. Ensures tailoring of software quality, safety, and verification and validation (V&V) requirements are based on software classification and safety level-of-effort.

f. Supports the CCB as a voting member for software-safety-related matters and impacts, and verifies implementation to configuration item(s).

g. Performs software quality activities, as defined in AFOP-7150.2-004, Software Assurance, including formal evaluations of process / plan compliance and verifications of product conformance.

h. Performs software safety activities, as defined in AFOP-7150.2-004, including identification of potential hazards associated with the software throughout the software development program as part of the SSWG.

2.6 Software Manager (SM)

The Flight Instrumentation & System Integration Branch Chief designates a qualified branch member to fulfill the position of SM on each project. The SM has the following roles and responsibilities:

a. Works to define the generation of the software classification for all configuration item(s), in conjunction with the SQA and SDA, the SM.

b. Defines the software development process, modification, maintenance, operations, retirement, management, acquisition, and assurance activities.

c. Establishes a reporting channel and interfaces with the software provider's project management that is independent of the software development function and the software assurance function.

d. Ensures the development requirements specified in this directive are addressed in the SMP / SDP, software test plan, and version description document (VDD) are met.

e. Identifies software configuration items, defines requirements, software or firmware development, and maintains all software configuration items, which also includes requirements tracing for the complete software life cycle, if required.

f. Identifies all COTS, Modified Off-The-Shelf (MOTS), and GOTS products for its capability to meet safety critical functions, interface to developed code, and the ability to verify at the same level as developed code.

g. Defines, traces, analyzes, and ensures compliance with all software requirements from one life cycle phase to another.

h. Establishes software configuration baselines and any changes to the baseline, ensuring the smooth transaction of software products from the development baseline to the project baseline.

i. Specifies the flight software to be flown on a designated flight using configuration change requests (CR), flight media releases, NASA Aircraft Management Information System, and work orders for on-aircraft software installations. This includes both informal and formal flight and flight support loads.

j. Serves as a voting member of the CCB for software-related matters.

k. Establishes and / or approves procedures for production of flight software media in the SMP / SDP.

l. Defines procedures for physically and electronically controlling flight software media as documented in the SDP and / or CMP.

m. Supports the identification, analysis, and / or generation of software hazards by participating in the SSWG.

n. Evaluates project tools for safety impact and, if necessary, documenting how project tools are selected, approved, and controlled.

o. Supports any NASA Headquarters' Independent Verification and Validation (IV&V) requirements, if applicable.

2.7 Configuration Manager

The Flight Instrumentation & Systems Integration Branch Chief designates a qualified branch member to fulfill the position of Configuration Manager on each project. The Configuration Manager has the following roles and responsibilities:

a. Establishes a system of software configuration identification of Center software products for all software classifications.

b. Provides a system for software configuration baseline management.

c. Provides a system for reviews and audits Center software products for Classification I, II, III, and S (see classification guidelines in this directive).

d. Provides a system of configuration status accounting for Center software products for all software classifications.

2.8 Flight Systems Lead (FSL)

a. Ensures that the system technically fulfills the defined needs and requirements and that the SE approach is being followed.

b. Oversees the project's engineering activities as performed by the Flight Instrumentation & System Integration Branch engineer.

c. Directs, communicates, monitors, and coordinates tasks, schedules, procurements, staffing levels, and necessary training.

d. Reviews and evaluates the technical aspects of the project to ensure that the systems / subsystems engineering processes are functioning properly and evolves the system from concept to product.

The entire technical team is involved in the SE process. The FSL focuses on the technical characteristics of decisions including technical, cost, and schedule, and on providing these to the PM and Chief Engineer (CE). The overlap in these responsibilities is natural, with the FSL and CE focused on the success of the engineering of the system (technical, cost, schedule) and the PM providing constraints on engineering options to maintain a successful delivery of the system within cost and schedule.

The FSL should play a key role in leading the development of the concept of operations, system architecture, defining boundaries, defining and allocating requirements, evaluating design tradeoffs, balancing technical risk between systems, defining and assessing interfaces, and providing oversight of V&V activities, as well as many other tasks.

The FSL should be involved in project-level planning, scheduling, and staffing exercises and developing higher level project documents. The FSL may provide input to the project level documents but is usually not responsible for preparing the documents.

Chapter 3: Software Classification

NPR 7150.2 defines six classifications for NASA software. These classifications are based on NPR 7150.2:

a. Usage with or within a NASA system,

b. Criticality of the system to NASA's major programs and projects,

c. Extent to which humans depend on the system,

d. Developmental and operational complexity, and

e. Extent of the Agency's investment.

These definitions are assigned an alphabetic classification identification from A-F. Classes A-E covers engineering software while Class F covers general purpose computing, business and IT software. In addition, NPR 7150.2 identifies an additional test applied to software defined as Class A, B, C, D, or E. This is the software safety criticality (reference NASA-STD-8739.8). The results of the safety-criticality test will change the number of NPR 7150.2 requirements levied on the software. In addition, it will levy the requirements defined in NASA-STD-8739.8 regarding safety and hazards. Historically, both the Center and the aeronautics industry have used a hazard- / risk-based software classification system that classifies software based on the effects of the failure of the software to function properly. Classification definitions used for commercial aircraft certification can be found in RTCA DO-178 C. The historical software classification system used at the Center is defined in AFOP-7150.2-004. The classification method used in this directive applies the hazard- / risk-based approach, commonly used on aeronautics-based platforms, while meeting the intent of the classification process found in NPR 7150.2, as it applies to aeronautics-based platforms. The scope of this directive only includes NPR 7150.2 Classes C, D, and E. The scope does not include NPR 7150.2 Class A, B, or F software requirements, since the Center does not currently generate this level of software. For software that is classified into NPR 7150.2 Class A, B, or F, please refer directly to NPR 7150.2.

3.1 General

Requirements in this directive are assigned to software items according to the criticality of that software. Specifically, software is grouped into one of four different classifications based on the most severe consequence of a software-controlled event. These classifications are closely coupled to the hazard categories described in AFG-7900.3-031. Specifically, these categories are as follows:

a. Class I: Catastrophic

b. Class II: Critical

c. Class III: Minor

d. Class IV: Negligible

The use of Roman Numerals is meant to reduce confusion with other software standards such as NPR 7150.2 and RTCA DO-178 C that use alphabetic classifications, and to align with the hazard severity classification in AFOP-7150.2-004. If more than one software class appears to apply, the higher of the classes is assigned to the system / subsystem.

Since this directive attempts to define both the engineering and business / IT software, the category definitions have been expanded to address other types of consequences, such as a software-related security breach or Agency-wide loss of productivity. For example, software that could cause a critical security breach would be classified as Class II.

Identification / incorporation of the safety-critical software increases the number of requirements levied by NPR 7150.2. It also levies additional requirements called out in NASA-STD-8739.8. Determination of the existence of safety-critical software involves performing a system / software safety assessment. If the system and software are determined to be safety critical, an additional “S” will be added to the classification to denote the presence of safety-critical software. For example, software that could cause a critical injury would be classified as Class II-S.

3.2 The Classification Process

In addition to the Safety Criticality Test, the criticality of a software item should also be determined using a PHA which is performed during system architectural development. The system level PHA will provide an initial assessment of the system / software hazards. From this, preliminary system / software level classifications can be determined. The PHA will be further refined as the software architecture matures until hazards have been reviewed down to the computer software configuration item (CSCI) level. Once this level is reached, the software CM system treats the software as a single entity.

If a CSCI has multiple categories of failures associated with its different functions, that item could be further partitioned to limit the interaction between software items. This may allow those items to be developed at different assurance levels, minimizing the volume of code that is to be developed to the more stringent standards.

For CSCIs that support multiple functions, the classification should be based on the most severe of the effects resulting from the failure or malfunction of any supported function or any combination of supported functions.

Analyze software design to ensure that any partitioning or isolation methods used in the design to logically isolate the safety-critical design elements from those that are non-safety are effective. Any software that can write or provide data to safety-critical software will also be considered safety critical, unless there is isolation built in, then the isolation design is considered safety critical. (See NASA-STD-8739.8, Software Assurance and Software Safety Standard).

Once the software is assessed to the CSCI level, perform a bottom-up review of the software architecture to ensure that CSCIs of differing classifications do not interact in such a way where the failure of a higher classification CSCI (i.e., Class IV) could cause a lower classification CSCI (i.e., Class I) to fail.

Note: As part of the initial PHA, the assessment should include a check to ensure that the vehicle / project does not fall into the large-scale aeronautics vehicle category. If it does, the program / project needs to consult with Center management to discuss the SWE approach to be used. (Exceeding a $250M total life-cycle cost results in software declared Class B per NPR 7150.2.)

3.3 Criteria for Safety Critical Software

Software is considered safety critical if it resides on a safety-critical system and meets the criteria defined in NASA-STD-8739.8 and AFOP-7150.2-004.

In accordance with AFPD-8700.1-001, safety programs are implemented for activities that are internally controlled by the Center or are operations sponsored or supported by the Center where:

a. Any NASA Center or its contractor personnel and its equipment are at risk,

b. The Center has an assigned safety responsibility (i.e., flight, ground, range, etc.),

c. Any NASA Center owns the asset and are not otherwise excluded by agreement or contract, or

d. A contractor owns the asset and is not otherwise excluded by agreement or contract.

This includes the following activities: aviation activity, project activity, and industrial activity. (See AFPD-8700.1-001 for definitions of these activities.)

3.4 Software Classification - Safety Critical

Software considered safety critical, using the definition in Appendix A of this directive, is further classified based on the most severe consequence of a software-controlled event. The classification criteria are as follows:

Note: Classification of software cannot be waived. Follow the process in AFPR-7123.2-001 to waive and deviate from Agency-defined requirements. See Appendix A of this directive for definitions of "Deviation" and "Waiver."

a. Class I-S: Catastrophic

(1) Death or permanently disabling / life-threatening injury.

(2) Destruction of facility on the ground, major system, vehicle, termination of project and mission failure of $2,000,000 or more.

b. Class II-S: Critical

(1) Severe / lost time injury or occupational illness.

(2) Major loss / damage to facility, system, equipment, flight hardware, vehicle, and mission failure of at least $500,000, but less than $2,000,000.

c. Class III-S *: Not Applicable

d. Class IV-S *: Not Applicable

**Note: A software component that is deemed to be safety-critical software, by definition, is a Software Safety Classification of Class I or Class II. Therefore, Class III-S and Class IV-S are not applicable.*

3.5 Software Classification – Non-Safety Critical

Software not considered safety critical, using the definition in Appendix A of this directive, is further classified based on the most severe consequence of a software-controlled event. The classification criteria are as follows:

a. Class I: Catastrophic

(1) Loss of the only opportunity for critical data.

b. Class II: Critical

(1) Long-term project delay.

(2) Loss of some project-critical data.

(3) Loss of confidentiality, integrity, and / or availability of information with an IT security category of "High" per NPR 2810.1, Security of Information and Information Systems.

(4) Agency-wide productivity impact.

c. Class III: Moderate

(1) Loss of mission (sortie, flight, return-to-base, test shut-down, etc.).

(2) Loss of non-critical project data.

(3) Loss of confidentiality, integrity, and / or availability of information with an IT security category of “Moderate”.

(4) Minor loss / damage to facility, system, equipment, or flight hardware and mission failure of at least $50,000, but less than $500,000.

(5) Interruptions in the availability of critical data.

(6) Center-wide productivity impact.

d. Class IV: Minimal

Productivity impact to small number of users.

Note: The monetary values for mission failure and property damage are found in AFRC 8621.1-001. The costs found in this directive should be used only as a reference. If a discrepancy exists between the specified recovery / replacement costs found in this directive, AFRC 8621.1-001 takes precedence.

3.6 Classification Guidelines

Software classification is not an exact science and is evaluated on a case-by-case basis. Some guidelines are given below:

a. Destruction of facility, major system, or vehicle

The intent of this statement is to capture consequences that would likely lead to a NASA Class A Mishap per NPR 8621.1, hull loss of a crewed aircraft or greater than $2,000,000 in property damage to a facility or system. However, in some cases, loss of a test article is either planned or anticipated and, thus, may not drive software criticality to the highest level. Examples include:

(1) Intentional destruction of a vehicle.

(2) Vehicles or systems not intended to be recovered once the test is complete.

b. Recovery / Replacement Costs

AFG-7900.3-031 and NPR 8621.1 both provide criteria for recovery / replacement costs (for instance, $2,000,000 is the threshold for a Category I Hazard in AFG-7900.3-031, and a Type A Mishap in NPR 8621.1). NPR 8621.1 provides guidance as to how to make that assessment.

c. Major Damage vs. Destruction

The distinction between major damage and destruction of a system should be determined by the feasibility of repair. If the system can be repaired within the cost and budget constraints of the project or program, it should be considered damaged. If repair is impossible or the costs prohibitive, it should be considered destroyed.

d. IT-Related Software

When software falls into the category of business and IT infrastructure, as defined in NPR 7150.2, then it should be classified in accordance with the guidance provided by the Center Chief Information Officer (CIO). The Center CIO may decide to apply the business and IT infrastructure software Classifications F, found in NPR 7150.2, in lieu of the classifications defined in this directive.

e. Long-Term Delay

The definition of long-term delay is project- or program-specific. A delay that constitutes some significant percentage of the project or program schedule (>5%) would certainly be considered a long-term delay. A delay that could trigger a high-level program review or project cancellation would also be considered long term.

f. Loss of Missions

Defining what constitutes a loss of mission is also highly program- or project-dependent. In some cases, mission and project are synonymous, and a failure to meet preapproved minimum mission success criteria indicates that project objectives were not met. This is the case where there is only one opportunity to gather the critical data. In other cases, loss of mission may imply loss of a single aircraft sortie, which has a much lower consequence. For the purposes of this directive, loss of mission implies that there will be other opportunities to collect the data.

g. Interruptions in Availability

An interruption in availability occurs when data, stored or real time, is not accessible. This could occur if the system used to access backed-up data fails, or if display software becomes inoperative. In those cases where real-time monitoring of data becomes impossible, other impacts may become the driver for criticality determination.

3.7 Software Classification Testing

Software testing is used to demonstrate that the software satisfies its requirements and to demonstrate with a high degree of confidence that errors that could lead to unacceptable failure conditions, as determined by the system safety assessment process, have been removed.

Test coverage analysis is a two-step process involving Requirements-Based and Structural-Based Coverage Analyses. The first step analyzed the test cases in relation to the software requirements to confirm that the selected test cases satisfy the specified criteria. The second step confirms that the requirements-based test procedures exercised the code structure to the applicable coverage criteria. Coverage analysis is the process of determining the degree to which a proposed software verification process activity satisfies its objectives.

The Requirements-Based Coverage Analysis test is used to determine how well the requirements-based testing verified the implementation of the software requirements. This analysis may reveal the need for additional requirements-based test cases.

The Structural-Based Coverage Analysis test determines which code structure, including interfaces between components, was not exercised by the requirements-based test procedures. The requirements-based test cases may not have completely exercised the code structure, including interfaces, so structural coverage analysis is performed, and additional verification produced to provide structural coverage. Structural-based testing may be performed on the Source Code, Object Code, or Executable Object Code.

The SMP / SDP will define the structural coverage testing required of the software based on the Software Classification. The structural coverage testing and the required percent coverage will be agreed upon by the Software ITA and acknowledged with their signature on the SMP / SDP.

The current software classifications and associated testing are as follows:

a. Class I / II: Modified Condition / Decision Coverage (MC / DC) (SWE-219)

Every point of entry and exit in a program has been invoked at least once, every condition in a decision in the program has taken all possible outcomes at least once, every decision in the program has taken all possible outcomes at least once, and each condition in a decision has been shown to independently affect that decision’s outcome. A condition is shown to independently affect a decision’s outcome by varying just that condition while holding fixed all other possible conditions or varying just that condition while holding fixed all other possible conditions that could affect the outcome (RTCA DO-178C).

b. Class I / II: Cyclomatic Complexity (SWE-220)

Safety Critical Software components have a cyclomatic complexity value of 15 or lower. Cyclomatic complexity is a metric used to measure the complexity of a software program. This metric measures independent paths through the source code. The point of this requirement is to minimize risk, minimize testing, and increase reliability associated with safety-critical software code components, thus reducing the chance of software failure during a hazardous event. The software developer assesses all software safety-critical components with a cyclomatic complexity score over 15 for testability, maintainability, and code quality. For more guidance on this requirement, see NASA-HDBK-2203.

c. Class III: Statement Coverage

Every statement in the program has been invoked at least once (RTCA DO-178C).

Note: Statement is as defined by the programming language.

3.8 Architectural Considerations

In some cases, mitigations to software hazards can be used to lower the classification of that software, if the following criteria are met:

a. The hazard has been mitigated through system design or through the use of safety devices (see Table 3-1.).

b. These mitigations meet the requirements levied in NPR 8715.3, Requesting Relief from Agency Mission Assurance Requirements.

c. These mitigations are verifiable and verified.

Note: Warning devices (i.e., a visual or audible alarm to the operator that a hazardous condition exists) or administrative / operational procedures (rules that limit use of the system to areas where the consequence of failure is more benign) alone cannot be used to reduce software classification.

Table 3-1: Architectural Considerations in Software Criticality Assessment

Mitigation Type	Description	Effect on Classification
Design	Other aspects of the system design (hardware or software) prevent the software from generating a hazardous condition (should be independent and not running on the same processor).	Can be considered when classifying the criticality of the software.

Mitigation Type	Description	Effect on Classification
Safety Devices	Other elements of the system identify and mitigate hazardous conditions before damage can occur.	Can be considered when classifying the criticality of the software.
Caution / Warning Devices	Other elements of the system that warn the operator if a hazardous condition is detected.	Should not be considered when classifying the criticality of the software.
Operational / Administrative Procedures	Rules regarding the operation or use of the system to limit the effects of hazardous conditions caused by software.	Should not be considered when classifying the criticality of the software.

Chapter 4: Implementation of SWE Requirements

This section describes the implementation of the requirements from NPR 7150.2 and NASA-STD-8739.8, as well as the Center-specific requirements. Requirements levied by NPR-7150.2 on the Center are captured in the first subsection. The subsequent subsections describe the project activities that are intended to meet the requirements from NASA-STD-8739.8 and NPR 7150.2 for each class of software.

Note: The scope of this directive only includes NPR 7150.2 Classes C, D, and E. The scope does not include NPR 7150.2 Class A, B, or F software requirements since the Center does not currently generate this level of software. For software that is classified into NPR 7150.2 Class A, B, or F, please refer directly to NPR 7150.2.

Table 4-1. Requirements Levied on the Center

NPR 7150.2 Requirement		Armstrong Implementation
	Requirements Levied on Center Directors or Designees	
SWE-003	... shall maintain, staff, and implement a plan to continually advance the Center's in-house SWE capability and monitor the SWE capability of NASA's contractors.	The TA and Software Working Group Lead will establish an informal list of improvements and review the list annually to check progress and relevance
SWE-005	... shall establish, document, execute, and maintain software processes.	AFPR (Armstrong Procedural Requirements (AFPR)), AFOP, Handbook, project records
SWE-006	... shall maintain a reliable list of their Center's programs and projects containing Class A, B, C, and D software. The list should include a. Project / program name and Work Breakdown Structure (WBS) number. b. Software name(s) and WBS number(s). c. Software size estimate (report in Kilo / Thousand Source Lines of Code (KSLOCs)). d. Phase of development or operations. e. Software Class or list of the software classes being developed on the project. f. Software Safety-Critical status g. For each CSCI / Major System containing Class A, B, or C software, provide: (1) The name of the software development organization. (2) Title or brief description of the CSCI / Major System. (3) The estimated total KSLOC the CSCI / Major System represents. (4) The primary programing languages used. (5) Primary life-cycle methodology being used on the software project. (6) Name of responsible software assurance organization(s).	The Center participates in the biennial Agency software inventory. The TA maintains a library of active SDPs and SMPs.
SWE-091	For Class C safety-critical software projects, … shall establish and maintain a software measurement repository for software project measurements containing at a minimum:	The Center will collect data on Class I and Class II projects that start coding after 1/1/2017.

NPR 7150.2 Requirement		Armstrong Implementation
	a. Software development tracking data b. Software functionality achieved data c. Software quality data d. Software development effort and cost data	
SWE-092	For Class C safety-critical software projects, … shall utilize software measurement data for monitoring SWE capability, improving software quality, and tracking the status of SWE improvement activities.	
SWE-095	... shall periodically report on the status of the Center's SWE discipline, as applied to its projects, upon request by the NASA Office of Chief Engineer, Office of Safety and Mission Assurance or Office of the Chief Health and Medical Officer.	The TA participates in capability leadership team and oversees Center participation in Software Working Group
SWE-140	... shall comply with the requirements in Appendix E, of this directive (NPR 7150.2), that are marked with an "X."	AFPR
SWE-142	For Class C software projects, ... shall establish and maintain a software cost repository(ies) that contains at least one of the following measures: a. Planned and actual effort and cost. b. Planned and actual schedule dates for major milestones. c. Both planned and actual values for key cost parameters that typically include software size, requirements count, defects counts for maintenance or sustaining engineering projects, and cost model inputs. d. Project descriptors or metadata that typically includes software class, software domain / type, and requirements volatility.	The Center will collect data on Class I and Class II projects that start coding after **1/1/2017**.
SWE-144	... shall contribute applicable SWE process assets in use at respective Centers to the Agency-wide process asset library.	The Center will contribute, as required.
SWE-150	…engineering, CIO, and S&MA authorities shall review and agree with any tailored NPR 7150.2 requirements per the requirements mapping matrix authority column.	Engineering and S&MA perform independent classification and coordinate for final tailoring
SWE-214	… shall perform the following actions for each type of internal NASA software transfer or reuse: a. A NASA civil servant to a NASA civil servant:	Release software via the Agency software release system (SRS) website: https://softwarerelease.ndc.nasa.gov/

NPR 7150.2 Requirement		Armstrong Implementation
	(1) Verify the requesting NASA civil servant has requested and completed an Acknowledgment (as set forth in the note following paragraph 3.10.2e in NPR 7150.2D). (2) Provide the software to the requesting NASA civil servant. b. A NASA civil servant to a NASA contractor: (1) Verify a NASA civil servant (e.g., a Contracting Officer (CO) or Contracting Officer Representative (COR)) has confirmed the NASA contractor requires such software for the performance of Government work under their NASA contract and that such performance of work will be a Government purpose. Center Intellectual Property Counsel should be consulted for any questions regarding what is or is not a Government purpose. (2) Verify a NASA civil servant (e.g., a CO or COR) has confirmed an appropriate Government Furnished Software clause (e.g., 1852.227-88, “Government-furnished computer software and related technical data”) is in the subject contract (or, if not, that such clause is first added); or the contractor may also obtain access to the software in accordance with the external release requirements of NPR 2210.1, Release of NASA Software. (3) Verify NASA contractor is not a foreign person (as defined by 22 Codes of Federal Regulation §120.16). (4) Verify there is a requesting NASA Civil servant (e.g., a CO or OR), and the requesting NASA civil servant has executed an Acknowledgment (as set forth in the note following paragraph 3.10.2e in NPR 7150.2D). (5) After items (1), (2), (3), and (4) are complete, provide the software to the requesting NASA civil servant. The requesting NASA civil servant is responsible for furnishing the software to the contractor pursuant to the subject contract’s terms. c. A NASA civil servant to any NASA grantees, Cooperative Agreement Recipients or any other agreement partners or to any other entity under UnitedStates (U.S.) Government Agency Release, Open source Release, Public Release, U.S. Release, Foreign Release:	

NPR 7150.2 Requirement		Armstrong Implementation
	(1) If the release is to any NASA grantees, Cooperative Agreement Recipients, or any other agreement (e.g., Space Act Agreement) partners or to any other entity under U.S. Government Agency Release, an Open source Release, a Public Release, a U.S. Release, or a Foreign Release, the software release is completed in accordance with the external release requirements of NPR 2210.1, Release of NASA Software – Revalidated w/ change 1.	
SWE-215	… shall ensure that the Government has clear rights in the software, a Government purpose license, or other appropriate license or permission from third party owners prior to providing the software for internal NASA software sharing or reuse.	Agency SRS website: https://softwarerelease.ndc.nasa.gov/
SWE-216	… shall ensure that all software listed on the internal software sharing or reuse catalog(s) conforms to NASA software engineering policy and requirements.	Agency SRS and NASA Software Catalog websites: https://softwarerelease.ndc.nasa.gov/ https://software.nasa.gov/
SWE-217	… shall perform the following actions: a. Keep a list of all contributors to the software product. b. Ensure that the software product contains appropriate disclaimer and indemnification provisions (e.g., in a “README” file) stating that the software may be subject to U.S. export control restrictions, and it is provided “as is” without any warranty, express or implied, and that the recipient waives any claims against, and indemnifies and holds harmless, NASA and its contractors and subcontractors.	Agency SRS website: https://softwarerelease.ndc.nasa.gov/
SWE-218	… shall ensure that the appropriate FAR, NFS, and other provisions / clauses based on this requirements document and NASA-STD-8739.8 are included for all NASA contracts, Space Act Agreements, cooperative agreements, partnership agreements, grants, or other agreements pursuant to which software is being acquired, developed, modified, operated, or managed for NASA.	Agency SRS and Submitting contracts / agreements to Contracts office and working with CO for the Project https://softwarerelease.ndc.nasa.gov/

NPR 7150.2 Requirement		Armstrong Implementation
	Requirements Levied on the Center Engineering TA	**Armstrong Implementation**
SWE-002	…shall lead, maintain, and fund a NASA SWE initiative to advance SWE practices.	NASA Headquarters funds the SWE Capability Leadership Team (CLT). Each Center has a current Center SWE Improvement Plan on file in the NASA Office of the CE.
SWE-004	…shall periodically benchmark each Center's SWE capability against requirements in NPR7150.2.	NASA Software Working Group and NASA CLT periodically benchmark each center's software capabilities.
SWE-021	… shall update their plan, if a system or subsystem development evolves to meet a higher or lower software classification as defined in Appendix D of NPR 7150.2, and initiate modifications to any supplier contracts to fulfill the applicable requirements per the Requirements Mapping Matrix in Appendix C of NPR 7150.2 and any approved tailoring.	Documented in each Projects SMP / SDP - see SWE-176
SWE-098	…shall maintain an Agency-wide process asset library of applicable best practices and process templates for all size projects	NASA Headquarters maintains process asset library.
SWE-100	…shall provide training to advance SWE practices.	SWE training
SWE-126	Serving as technical and institutional authorities for requirements in this directive…shall: Assess projects' requirements mapping matrices and tailoring from requirements in this directive by: 1. Checking the accuracy of the project's classification of software components against the definitions in NPR 7150.2. 2. Evaluating the project's Requirements Mapping Matrix for commitments to meet applicable requirements in this directive, consistent with software classification. 3. Confirming that requirements marked "Not Applicable" in the project's Requirements Mapping matrix are not relevant or not capable of being applied.	TA signature on project development and management plans

NPR 7150.2 Requirement		Armstrong Implementation
	4. Determining whether the project's risks, mitigations, and related requests for relief from requirements designated with "X" in NPR 7150.2 are reasonable and acceptable. 5. Approving / disapproving requests for relief from requirements designated with "X" in Appendix C of NPR 7150.2D, which fall under this TA's scope of responsibility. 6. Facilitating the processing of projects' requirements mapping matrices and tailoring decisions from requirements in this directive, which fall under the responsibilities of a different TA (See NPR 7150.2 Appendix C Column titled “Authority”). 7. Include the Senior Agency Information Security Officer (or delegate) in all software reviews to ensure software cybersecurity is included throughout software development, testing, maintenance, retirement, operations, management, acquisition, and assurance activities 8. Ensuring that approved requirements mapping matrices, including any tailoring rationale against this directive, are archived as part of retrievable project records.	Waivers are processed using the Center process per AFPR-7123.2-001.
SWE-129	… shall authorize appraisals against selected requirements in NPR 7150.2 to check compliance	NASA Headquarters authorizes appraisals.
SWE-152	… shall periodically review project requirements mapping matrices.	A software compliance matrix audit checklist added to all future SMPs / SDPs. The audit will be performed by the Software ITA, or delegate, throughout the life cycle of the software. The completed checklist is to be retained in project repository.
SWE-153	... shall define the content requirements for software documents or records.	Defined in SWEHB
SWE-208	… shall lead and maintain a NASA Software Assurance and Software Safety Initiative to advance software assurance and software safety practices	See Center Chief S&MA
SWE-209	… shall periodically benchmark each Center’s software assurance and software safety capabilities against the NASA-STD-8739.8, NASA Software Assurance and Software Safety Standard.	See Center Chief S&MA
SWE-212	… shall periodically review the project’s requirements mapping matrices.	Documented in each Projects SMP / SDP via approval and signature cycle / review

NPR 7150.2 Requirement		Armstrong Implementation
SWE-221	… shall authorize appraisals against selected requirements in this NPR to check compliance.	See Center Chief S&MA
SWE-222	… shall provide for software assurance training.	See Center Chief S&MA
SWE-223	… shall make the final decision on all proposed tailoring of SWE-141, the IV&V requirement.	Non-Applicable - all Armstrong development in Class A / B will follow NPR

Table 4-2. Class IV Software Implementation

SWE numbers marked with an asterisk (*) are required for ***FLIGHT*** software only.

Step	Activity	Applicable SWE Requirements		Waivable by Center Software ITA?
1	Write down what your software is supposed to do.	SWE-050*	… shall establish, capture, record, approve, and maintain software requirements, including requirements for COTS, GOTS, MOTS, Open Source Software (OSS), or reused software components, as part of the technical specification. Flight only – Exclude COTS / GOTS / MOTS	Yes
2	When the purpose of your software changes, update.	SWE-053*	...shall track and manage changes to the software requirements Flight only – Exclude COTS / GOTS / MOTS	Yes
3	Use version control (subversion, Git, etc.) as you develop and test.	SWE-080*	... shall track and evaluate changes to software products. Flight Only – exclude COTS / GOTS / MOTS	Yes
		SWE-081*	... shall identify the software configuration items (e.g., software records, code, data, tools, models, scripts) and their versions to be controlled for the project. Flight only – exclude COTS	Yes
4	Write simple test plans, perform software test, and record results and track defects	SWE-066*	... shall perform software testing. Flight only – Exclude COTS / GOTS / MOTS	Yes
5	Acquisitions activities	SWE-033	… shall assess options for software acquisition versus development.	Yes
6	Complete a compliance matrix and get the TA to sign it.	SWE-013	... shall develop, maintain, and execute software plans, including security plans, that cover the entire software life cycle and, as a minimum, address the requirements of this directive with approved tailoring.	Yes
		SWE-125	… shall [with software components] maintain a requirements mapping matrix or multiple requirements mapping matrices against requirements in this NPR,	Yes

			including those delegated to other parties or accomplished by contract vehicles or Space Act Agreements.	
		SWE-139	… shall comply with the requirements in this NPR [7150.2] that are marked with an ”X” in Appendix C consistent with their software classification.	Yes
7	Evaluate software for potential reuse and contribute reuse candidates to Agency Software Catalog.	SWE-148	… shall evaluate software for potential reuse by other projects across NASA and contribute reuse candidates to the NASA Internal Sharing and Reuse Software systems. However, if the project manager is a contractor, then a civil servant needs to pre-approve all such software contributions; all software contributions should include, at a minimum, the following information: a. Software Title. b. Software Description. c. The Civil Servant Software Technical Point of Contact for the software product. d. The language or languages used to develop the software. Any third-party code contained therein, and the record of the requisite license or permission received from the third party permitting the Government’s use and any required markings (e.g., required copyright, author, applicable license notices within the software code, and the source of each third-party software component (e.g., software uniform resource locator (URL) & license URL)), if applicable. e. Release notes.	Yes
8	Work with Flight Instrumentation & Systems Integration Branch Chief to identify requisite skills for the software effort and provide training, as appropriate.	SWE-121	[Where approved,] … shall document and reflect the tailored requirement in the plans or procedures controlling the development, acquisition, and deployment of the affected software.	Yes
9	Classify the software per this directive	SWE-020	... shall classify each system and subsystem containing software in accordance with the highest applicable software classification definitions for ...	Yes
		SWE-176	… shall maintain records of each software classification determination, each software Requirements Mapping Matrix, and the results of each software independent classification assessments for the life of the project.	Yes

		SWE-205	… shall, in conjunction with the S&MA organization, determine if each software component is considered to be safety-critical per the criteria defined in NASA-STD-8739.8.	ITA / S&MA
10	Require the software developer to provide NASA with electronic access to source code in modifiable format, including MOTS software, non-flight software.	SWE-042	… shall require the software developer(s) to provide NASA with electronic access to the source code developed for the project in a modifiable format.	Yes
11	Establish and maintain requirements, test plans / procedures and results	SWE-065*	…shall establish and maintain: a. Software test plan(s). b. Software test procedure(s). c. Software test(s), including any code specifically written to perform test procedures. d. Software test report(s). Flight and External Releases only	Yes
		SWE-071*	… shall update software test plan(s) and software test procedures(s) to be consistent with software requirements Flight and External Releases only	Yes
12	Plan for software assurance activities.	SWE-022	… shall plan and implement software assurance per NASA-STD-8739.8.	Yes
13	Software Cybersecurity / Work with Project Cybersecurity Analyst	SWE-156	… shall perform a software cybersecurity assessment on the software components per the Agency security policies and the project requirements, including risks posed by the use of COTS, GOTS, MOTS, OSS, or reused software components.	ITA / Cybersecurity

Complete the following steps for software to be installed on a flight vehicle:				
		SWE-063*	... shall provide a software version description for each software release. [Flight and External releases only – exclude COTS]	Yes
14	Write a VDD that includes: a. What the software is supposed to do b. How to install the software c. How to operate the software d. List of software components and version numbers e. List of incorporated and outstanding bug fixes	AFOP-7150.2-004* (Flight Software Media Control 1)	Prior to installation on an aircraft, a VDD will be produced containing a form AFRC 80184, Flight Media Release, and attached a form AFRC 70010, Configuration Change Request, requesting installation. [Flight and External releases only – exclude COTS]	Yes
		AFOP-7150.2-004* (Flight Software Media Control 3)	…form AFRC 80184 that uniquely identifies (via checksum(s), file size / modification dates, or other verifiable means) the specific software load that should be installed on the aircraft	Yes
15	Prepare the form AFRC 80184	AFOP-7150.2-004* (Flight Software Media Control 4)	Flight software for a specific flight or block of flights … on a form AFRC 80184	Yes
		AFOP-7150.2-004* (Flight Software Media Control 5)	A procedure … written for flight software installation into the aircraft computer and for verification of correct loading.	Yes
16	Prepare the software installation procedure	AFOP-7150.2-004* (Flight Software Media Control 6)	Quality inspection shall verify the correct flight software is loaded for the specified flight according to approved procedures	Yes

Table 4-3. Class III Software (Mission-Critical, *Not Safety-Critical*) Implementation

Step	Activity	Applicable SWE Requirements		Waivable by Center Software ITA?
1	Classify the software per this directive	SWE-020	... shall classify each system and subsystem containing software in accordance with the highest applicable software classification definitions for NPR 7150.2 Classes A, B, C, D, E, F, G, and H software …	Yes
		SWE-176	… shall maintain records of each software classification determination, each software Requirements Mapping Matrix, and the results of each software independent classification assessments for the life of the project.	Yes
		SWE-205	… shall, in conjunction with the S&MA organization, determine if each software component is considered to be safety-critical per the criteria defined in NASA-STD-8739.8	ITA / S&MA
2	Project Schedule	SWE-016	… shall document and maintain a software schedule that satisfies the following conditions: a. Coordinates with the overall project schedule. b. Documents the interactions of milestones and deliverables between software, hardware, operations, and the rest of the system. c. Reflects the critical dependencies for software development activities. d. Identifies and accounts for dependencies with other projects and cross-program dependencies	Yes
		SWE-024	… shall track the actual results and performance of software activities against the software plans. a. Corrective actions are taken, recorded, and managed to closure. b. Including changes to commitments (e.g., software plans) that have been agreed to by the affected groups and individuals.	Yes
3	Project Cost Estimates	SWE-015	…To better estimate the cost of development, […] shall establish, document, and maintain: […] c. One software cost estimate model and associated cost parameter(s) for all C and D software projects.	Yes

<table>
<tr><td></td><td></td><td>SWE-151</td><td>… shall software cost estimate(s) satisfy the following conditions:
a. Covers the entire software life cycle.
b. Is based on selected project attributes (e.g., assessment of the size, functionality, complexity, criticality, reuse code, modified code, and risk of the software processes and products).
c. Is based on the cost implications of the technology to be used and the required maturation of technology.
d. Incorporates risk and uncertainty, including end state risk and threat assessment for cybersecurity.
e. Includes the cost of the required software assurance support.
f. Includes other direct costs.</td><td>Yes</td></tr>
<tr><td></td><td></td><td>SWE-174</td><td>… shall submit software planning parameters, including size and effort estimates, milestones, and characteristics, to the Center measurement repository at the conclusion of major milestones.</td><td>Yes</td></tr>
<tr><td>4</td><td>V&V Planning and Tracking
[Following Center V&V process will comply with NPR 7150.2 requirements]</td><td>SWE-068</td><td>… shall evaluate test results and record the evaluation.</td><td>Yes</td></tr>
<tr><td>5</td><td>Acceptance Criteria</td><td>SWE-034</td><td>... shall define and document the acceptance criteria for the software.</td><td>Yes</td></tr>
<tr><td></td><td></td><td>SWE-050</td><td>… shall establish, capture, record, approve, and maintain software requirements, including requirements for COTS, GOTS, MOTS, OSS, or reused software components, as part of the technical specification.</td><td>Yes</td></tr>
<tr><td>6</td><td>Requirements documents

[Note: Following Center requirements management process will comply with NPR 7150.2 requirements]</td><td>SWE-052</td><td>… shall perform, record, and maintain bidirectional traceability between the software requirement and the higher-level requirement.
<table>
<tr><th>Bi-directional Traceability</th><th>Class A, B, and C</th><th>Class D</th></tr>
<tr><td>Higher-level requirements to the software requirements</td><td>X</td><td></td></tr>
<tr><td>Software requirements to the system hazards</td><td>X</td><td>X</td></tr>
<tr><td>Software requirements to the software design components</td><td>X</td><td></td></tr>
<tr><td>Software design components to the software code</td><td>X</td><td></td></tr>
<tr><td>Software requirements to the software verification(s)</td><td>X</td><td>X</td></tr>
<tr><td>Software requirements to the software non-conformances</td><td>X</td><td>X</td></tr>
</table></td><td>Yes</td></tr>
</table>

		SWE-053	… shall track and manage changes to the software requirements.	Yes
7	Design Documents	SWE-065	... shall establish and maintain: a. Software test plan(s). b. Software test procedure(s). c. Software test(s), including any code specifically written to perform test procedures d. Software test report(s).	Yes
8	Test Plans and Procedures, Test Reports	SWE-071	... shall update software test plan(s) and software test procedure(s) to be consistent with software requirements.	Yes
9	Operations, Maintenance, Disposal Plans	SWE-075	... shall plan and implement software operations, maintenance, and retirement activities.	Yes
		SWE-195	… shall maintain the software using standards and processes per the applicable software classification throughout the maintenance phase.	Yes
		SWE-196	… shall identify the records and software tools to be archived, the location of the archive, and procedures for access to the products for software retirement or disposal.	Yes
10	Product Delivery Plans	SWE-077	... shall complete and deliver the software product to the customer with appropriate records, including as-built records, to support the operations and maintenance phase of the software's life cycle. [Applicable to External vendors only]	Yes
		SWE-085	... shall establish and implement procedures for the storage, handling, delivery, release, and maintenance of deliverable software products.	Yes
		SWE-194	… shall complete, prior to delivery, verification that all software requirements identified for this delivery have been met or dispositioned, that all approved changes have been implemented, and that all defects designated for resolution prior to delivery have been resolved.	Yes
11	Configuration Control Plans and Activities	SWE-079	... shall develop a software CMP that describes the functions, responsibilities, and authority for the implementation of software CM for the project.	Yes
		SWE-080	... shall track and evaluate changes to software products.	Yes

		SWE-081	... shall identify the software configuration items (e.g., software records, code, data, tools, models, scripts) and their versions to be controlled for the project.	Yes
12	Acquisitions activities	SWE-033	… shall assess options for software acquisition versus development.	Yes
13	Work with Flight Instrumentation & Systems Integration Branch Chief to identify requisite skills for the software effort and provide training, as appropriate.	SWE-121	[Where approved,] … shall document and reflect the tailored requirement in the plans or procedures controlling the development, acquisition, and deployment of the affected software.	Yes
14	Complete a compliance matrix and get the TA to sign it.	SWE-013	... shall develop, maintain, and execute software plans, including security plans, that cover the entire software life cycle and, as a minimum, address the requirements of this directive with approved tailoring.	Yes
		SWE-125	… [with software components] shall maintain a requirements mapping matrix or multiple requirements mapping matrices against requirements in this NPR, including those delegated to other parties or accomplished by contract vehicles or Space Act Agreements.	Yes
		SWE-139	… shall comply with the requirements in this NPR [7150.2D] that are marked with an ”X” in Appendix C consistent with their software classification.	Yes
16	Plan for software assurance activities.	SWE-022	… shall plan and implement software assurance per NASA-STD-8739.8. [CAP053:027]	Yes
17	Specify re-usability requirements that apply to software development activities to enable future reuse of the software, including models used to generate the software.	SWE-147	… shall specify reusability requirements that apply to its software development activities to enable future reuse of the software, including the models, simulations, and associated data used as inputs for auto-generation of software, for United States Government purposes. [as required]	Yes
18	Determine which processes, documents, electronic products, activities, and tasks are required.	SWE-036	… shall establish and maintain the software processes, software documentation plans, list of developed electronic products, deliverables, and list of tasks for the software development that are required for the project’s software developers, as well as the action required (e.g., approval, review) of the Government upon receipt of each of the deliverables.	Yes

19	Define milestones at which progress will be reviewed and audited.	SWE-037	… shall define and document the milestones at which the software developer(s) progress will be reviewed and audited.	Yes
20	Require supplier to allow NASA to: a. Monitor product integration b. Review verification activities c. Review trade studies and source data d. Audit software development process e. Participate in software reviews and Technical Interchange Meetings (TIMs)	SWE-039	… shall require the software developer(s) to periodically report status and provide insight into software development and test activities; at a minimum, the software developer(s) will be required to allow the project manager and software assurance personnel to: a. Monitor product integration. b. Review the verification activities to ensure adequacy. c. Review trades studies and source data. d. Audit the software development processes and practices. e. Participate in software reviews and technical interchange meetings.	Yes
21	Require the software developers to provide NASA with software products and process tracking information in electronic format, including software development and management metrics. The number of software defects assessed against mission success risks for proper resolution (vs. total number of software defects) can be used to meet the intent of software quality metric requirements to collect, analyze, and report out to the project per Class I and Class II. [CAP053:023]	SWE-040	… shall require the software developer(s) to provide NASA with software products, traceability, software change tracking information and nonconformances, in electronic format, including software development and management metrics.	Yes
22	Require the software developer to provide NASA with electronic access to source code in modifiable format, including MOTS software, non-flight software.	SWE-042	… shall require the software developer(s) to provide NASA with electronic access to the source code developed for the project in a modifiable format.	Yes
23	Require supplier to provide schedule and schedule updates, as needed.	SWE-046	... shall require the software developer(s) to provide a software schedule for the project's review and schedule updates as requested.	Yes

24	Evaluate software for potential reuse and contribute reuse candidates to Agency Software Catalog	SWE-148	… shall evaluate software for potential reuse by other projects across NASA and contribute reuse candidates to the NASA Internal Sharing and Reuse Software systems. However, if the project manager is a contractor, then a civil servant needs to pre-approve all such software contributions; all software contributions should include, at a minimum, the following information: a. Software Title. b. Software Description. c. The Civil Servant Software Technical Point of Contact for the software product. d. The language or languages used to develop the software. Any third-party code contained therein, and the record of the requisite license or permission received from the third party permitting the Government's use and any required markings (e.g., required copyright, author, applicable license notices within the software code, and the source of each third-party software component (e.g., software URL & license URL)), if applicable. e. Release notes.	Yes
25	Plan for requirements, documentation, rights, support, V&V, and vendor defect reporting of COTS, GOTS, MOTS, or reused software components.	SWE-027	... shall satisfy the following conditions when a COTS, GOTS, MOTS, or reused software component is acquired or used: a. The requirements to be met by the software component are identified. b. The software component includes documentation to fulfill its intended purpose (e.g., usage instructions). c. Proprietary rights, usage rights, ownership, warranty, licensing rights, and transfer rights have been addressed. d. Future support for the software product is planned and adequate for project needs. e. The software component is verified and validated to the same level required to accept a similar developed software component for its intended use. f. The project has a plan to perform periodic assessments of vendor reported defects to ensure the defects do not impact the selected software components.	Yes
26	Unit Test	SWE-062	… shall unit test the software code.	Yes
		SWE-186	… shall assure that the unit test results are repeatable.	Yes

		SWE-066	... shall test the software against its requirements.	Yes
		SWE-187	… shall place software items under configuration management prior to testing. *[post baseline only]*	Yes
		SWE-189	… shall ensure that the code coverage measurements for the software are selected, implemented, tracked, recorded, and reported. *Note*: *Percent coverage value is agreed and signed off by the Center's Engineering TA and measured*	Yes
27	Software Test / Code Statement Coverage	SWE-192	… shall verify through test the software requirements that trace to a hazardous event, cause, or mitigation technique.	ITA / S&MA
28	Implement mission – critical requirements from the NPR 7150.2	SWE-134	…, if a project has safety-critical software or mission-critical software, shall implement the following items in the software: a. The software is initialized, at first start and restarts, to a known safe state. b. The software safely transitions between all predefined known states. c. Termination performed by software of functions is performed to a known safe state. d. Operator overrides of software functions require at least two independent actions by an operator. e. Software rejects commands received out of sequence when execution of those commands out of sequence can cause a hazard. f. The software detects inadvertent memory modification and recovers to a known safe state. g. The software performs integrity checks on inputs and outputs to / from the software system. h. The software performs prerequisite checks prior to the execution of safety-critical software commands. i. No single software event or action is allowed to initiate an identified hazard. j. The software responds to an off-nominal condition within the time needed to prevent a hazardous event. k. The software provides error handling. l. The software can place the system into a safe state. Consult with Project Software Manager or ITA	Yes

29	Software Cybersecurity / Work with Project Cybersecurity Analyst	SWE-154	… shall identify cybersecurity risks, along with their mitigations, in flight and ground software systems and plan the mitigations for these systems.	ITA / Cybersecurity
		SWE-156	… shall perform a software cybersecurity assessment on the software components per the Agency security policies and the project requirements, including risks posed by the use of COTS, GOTS, MOTS, OSS, or reused software components.	ITA / Cybersecurity
		SWE-157	… shall implement protections for software systems with communications capabilities against unauthorized access per the requirements contained in the NASA-STD-1006, Space System Protection Standard.	ITA / Cybersecurity
		SWE-159	… shall test the software and record test results for the required software cybersecurity mitigation implementations identified from the security vulnerabilities and security weaknesses analysis.	ITA / Cybersecurity
30	Software Non-Conformance or Defect Management	SWE-201	… shall track and maintain software non-conformances (including defects in tools and appropriate ground software).	ITA / S&MA
31	Write a VDD that includes: a. What the software is supposed to do b. How to install the software c. How to operate the software d. List of software components and version numbers e. List of incorporated and outstanding bug fixes	SWE-063	…shall provide a software version description for each software release.	Yes
		AFOP-7150.2-004 (Flight Software Media Control 1)	Prior to installation on an aircraft, a VDD will be produced containing a form AFRC 80184 and an attached a form AFRC 70010 requesting installation.	Yes
32	Prepare software media, as required, for installation.	AFOP-7150.2-004 (Flight Software Media Control 2)	All software media (tape, disk or chip) … identified and physically labeled at the time of production. [Exclude – COT / GOTS]	Yes
33	Prepare the form AFRC 80184.	AFOP-7150.2-004 (Flight Software Media Control 3)	…form AFRC 80184 that uniquely identifies (via checksum(s), file size / modification dates, or other verifiable means) the specific software load that should be installed on the aircraft.	Yes

		AFOP-7150.2-004 (Flight Software Media Control 4)	Flight software for a specific flight or block of flights … on a form AFRC 80184.	Yes
34	Prepare and execute the software installation procedure.	AFOP-7150.2-004 (Flight Software Media Control 5)	A procedure … for flight software installation into the aircraft computer and for verification of correct loading.	Yes
		AFOP-7150.2-004 (Flight Software Media Control 6)	Quality inspection shall verify the correct flight software is loaded for the specified flight according to approved procedures.	Yes

Table 4-4. Class I/II (Safety and Non-Safety Critical) Combined

Step	Activity	Applicable SWE Requirements		Waivable by Center Software ITA?
1	Classify the software per this directive	SWE-020	... shall classify each system and subsystem containing software in accordance with the highest applicable software classification definitions for Classes A, B, C, D, E, F, G, and H software in NPR 7150.2.	Yes
		SWE-176	… shall maintain records of each software classification determination, each software Requirements Mapping Matrix, and the results of each software independent classification assessments for the life of the project.	Yes
		SWE-205	… shall, in conjunction with the S&MA organization, determine if each software component is considered to be safety-critical per the criteria defined in NASA-STD-8739.8.	ITA / S&MA
2	Project Schedule	SWE-016	… shall document and maintain a software schedule that satisfies the following conditions: a. Coordinates with the overall project schedule. b. Documents the interactions of milestones and deliverables between software, hardware, operations, and the rest of the system. c. Reflects the critical dependencies for software development activities. d. Identifies and accounts for dependencies with other projects and cross-program dependencies	Yes
		SWE-024	… shall track the actual results and performance of software activities against the software plans. a. Corrective actions are taken, recorded, and managed to closure. b. Including changes to commitments (e.g., software plans) that have been agreed to by the affected groups and individuals.	Yes
3	Project Cost Estimates	SWE-015	…To better estimate the cost of development, […] shall establish, document, and maintain: […] c. One software cost estimate model and associated cost parameter(s) for all C and D software projects.	Yes
		SWE-151	… shall software cost estimate(s) satisfy the following conditions: a. Covers the entire software life cycle. b. Is based on selected project attributes (e.g., assessment of the size, functionality, complexity, criticality, reuse code, modified code, and risk of the software processes and products).	Yes

			c. Is based on the cost implications of the technology to be used and the required maturation of technology. d. Incorporates risk and uncertainty, including end state risk and threat assessment for cybersecurity. e. Includes the cost of the required software assurance support. f. Includes other direct costs.	
		SWE-174	… shall submit software planning parameters, including size and effort estimates, milestones, and characteristics, to the Center measurement repository at the conclusion of major milestones.	Yes
4	Project Risk Management System	SWE-086	… shall record, analyze, plan, track, control, and communicate all of the software risks and mitigation plans.	Yes
5	Project Reviews / Status Activities	SWE-093	… shall analyze software measurement data collected using documented project-specified and / or Center / organizational analysis procedures.	Yes
		SWE-094	… shall provide access to the software measurement data, measurement analyses, and software development status as requested to the sponsoring Mission Directorate, the NASA CE, the Center Technical Authorities, Headquarters S&MA and other organizations as appropriate.	Yes
		SWE-199	… shall monitor measures to ensure the software will meet or exceed performance and functionality requirements, including satisfying constraints.	Yes
		SWE-018	… shall regularly hold reviews of software activities, status, and results with the project stakeholders and track issues to resolution.	Yes
		SWE-087	... shall perform and report the results of software peer reviews or software inspections for: 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 SDPs. d. Software code as defined in the software and / or project plans. e. Software test procedures.	Yes
		SWE-088	… shall, for each planned software peer review or software inspection: 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.	Yes

<table>
<tr><td></td><td></td><td></td><td>c. Track actions identified in the reviews until they are resolved.
d. Identify the required participants.</td><td></td></tr>
<tr><td></td><td></td><td>SWE-089</td><td>... shall for each planned software peer review or software inspection, record basic measurements.</td><td>Yes</td></tr>
<tr><td></td><td></td><td>SWE-090</td><td>... shall establish, record, maintain, report, and utilize software management and technical measurements.</td><td>Yes</td></tr>
<tr><td rowspan="2">6</td><td rowspan="2">V&V Planning and Tracking</td><td>SWE-055</td><td>… shall perform requirements validation to ensure that the software will perform as intended in the customer environment.</td><td>Yes</td></tr>
<tr><td>SWE-068</td><td>… shall evaluate test results and record the evaluation.</td><td>Yes</td></tr>
<tr><td>7</td><td>Acceptance Criteria</td><td>SWE-034</td><td>... shall define and document the acceptance criteria for the software.</td><td>Yes</td></tr>
<tr><td rowspan="6">8</td><td rowspan="6">Requirements Documents

Note: Following Center requirements management process will comply with NPR 7150.2 requirements.</td><td>SWE-050</td><td>… shall establish, capture, record, approve, and maintain software requirements, including requirements for COTS, GOTS, MOTS, OSS, or reused software components, as part of the technical specification.</td><td>Yes</td></tr>
<tr><td>SWE-051</td><td>… shall perform software requirements analysis based on flowed-down and derived requirements from the top-level SE requirements and the hardware specifications and design.</td><td>Yes</td></tr>
<tr><td>SWE-052</td><td>… shall perform, record, and maintain bi-directional traceability between the software requirement and the higher-level requirement.

<table>
<tr><th>Bi-directional Traceability</th><th>Class A, B, and C</th><th>Class D</th></tr>
<tr><td>Higher-level requirements to the software requirements</td><td>X</td><td></td></tr>
<tr><td>Software requirements to the system hazards</td><td>X</td><td>X</td></tr>
<tr><td>Software requirements to the software design components</td><td>X</td><td></td></tr>
<tr><td>Software design components to the software code</td><td>X</td><td></td></tr>
<tr><td>Software requirements to the software verification(s)</td><td>X</td><td>X</td></tr>
<tr><td>Software requirements to the software non-conformances</td><td>X</td><td>X</td></tr>
</table></td><td>Yes</td></tr>
<tr><td>SWE-053</td><td>… shall track and manage changes to the software requirements.</td><td>Yes</td></tr>
<tr><td>SWE-054</td><td>… shall identify, initiate corrective actions, and track until closure inconsistencies among requirements, project plans, and software products.</td><td>Yes</td></tr>
</table>

		SWE-184	… shall include software related safety constraints, controls, mitigations and assumptions between the hardware, operator, and software in the software requirements documentation.	ITA / S&MA
9	Test Plans and Procedures, Test Reports	SWE-065	... shall establish and maintain: a. Software test plan(s). b. Software test procedure(s). c. Software test(s), including any code specifically written to perform test procedures. d. Software test report(s).	Yes
		SWE-071	... shall update software test plan(s) and software test procedure(s) to be consistent with software requirements.	Yes
10	Operations, Maintenance, Disposal Plans	SWE-075	... shall plan and implement software operations, maintenance, and retirement activities.	Yes
		SWE-195	… shall maintain the software using standards and processes per the applicable software classification throughout the maintenance phase.	Yes
		SWE-196	… shall identify the records and software tools to be archived, the location of the archive, and procedures for access to the products for software retirement or disposal.	Yes
11	Product Delivery	SWE-077	... shall complete and deliver the software product to the customer with appropriate records, including as-built records, to support the operations and maintenance phase of the software's life cycle.	Yes
		SWE-085	... shall establish and implement procedures for the storage, handling, delivery, release, and maintenance of deliverable software products.	Yes
		SWE-194	… shall complete, prior to delivery, verification that all software requirements identified for this delivery have been met or dispositioned, that all approved changes have been implemented, and that all defects designated for resolution prior to delivery have been resolved.	Yes
12	Configuration Control Plans and Activities	SWE-079	... shall develop a software CMP that describes the functions, responsibilities, and authority for the implementation of software CM for the project.	Yes
		SWE-080	... shall track and evaluate changes to software products.	Yes
		SWE-081	... shall identify the software configuration items (e.g., software records, code, data, tools, models, scripts) and their versions to be controlled for the project.	Yes
		SWE-082	... shall establish and implement procedures to:	Yes

			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.	
		SWE-083	... shall prepare and maintain records of the configuration status of software configuration items.	Yes
		SWE-084	... 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.	Yes
13	Acquisition Activities	SWE-033	… shall assess options for software acquisition versus development.	Yes
14	Complete a compliance matrix and get the TA to sign it.	SWE-013	... shall develop, maintain, and execute software plans, including security plans, that cover the entire software life cycle and, as a minimum, address the requirements of this directive with approved tailoring.	Yes
		SWE-121	[Where approved,] … shall document and reflect the tailored requirement in the plans or procedures controlling the development, acquisition, and deployment of the affected software.	Yes
		SWE-125	… [with software components] shall maintain a requirements mapping matrix or multiple requirements mapping matrices against requirements in this NPR, including those delegated to other parties or accomplished by contract vehicles or Space Act Agreements.	Yes
		SWE-139	… shall comply with the requirements in this NPR [7150.2] that are marked with an ”X” in Appendix C consistent with their software classification.	Yes
15	Work with Flight Instrumentation & Systems Integration Branch Chief to identify requisite skills for the software effort and provide training, as appropriate.	SWE-017	... shall plan, track, and ensure project-specific software training for project personnel.	Yes
16	Develop and document the software architecture.	SWE-057	… shall transform the requirements for the software into a recorded software architecture.	Yes
17	Software Architecture Review	SWE-143	… shall perform a software architecture review on the following categories of projects: a. Category 1 Projects as defined in NPR 7120.5.	Yes

			b. Category 2 Projects as defined in NPR 7120.5 that have Class A or Class B payload risk classification per NPR 8705.4.	
18	Plan for coding, compilation, and testing of lower-level units of the software architecture.	SWE-058	... shall develop, record, and maintain a design based on the software architectural design that describes the lower-level units so that the units can be coded, compiled, and tested.	Yes
19	Select and verify adherence to software coding methods, standards, and / or criteria.	SWE-061	... shall select and adhere to software coding methods, standards, and criteria.	Yes
20	Validate and accredit software tools required to develop or maintain software. For safety-critical software development and evaluate tools for impact on system safety.	SWE-136	... shall validate and accredit software tool(s) required to develop or maintain software.	Yes
		SWE-070	... shall use validated and accredited software models, simulations, and analysis tools required to perform qualification of flight software or flight equipment.	Yes
21	Implement safety-critical OR mission – critical requirements from the NPR 7150.2.	SWE-134	…, if a project has safety-critical software or mission-critical software, shall implement the following items in the software: a. The software is initialized, at first start and restarts, to a known safe state. b. The software safely transitions between all predefined known states. c. Termination performed by software of functions is performed to a known safe state. d. Operator overrides of software functions require at least two independent actions by an operator. e. Software rejects commands received out of sequence when execution of those commands out of sequence can cause a hazard. f. The software detects inadvertent memory modification and recovers to a known safe state. g. The software performs integrity checks on inputs and outputs to / from the software system. h. The software performs prerequisite checks prior to the execution of safety-critical software commands. i. No single software event or action is allowed to initiate an identified hazard. j. The software responds to an off-nominal condition within the time needed to prevent a hazardous event.	Yes

			k. The software provides error handling. l. The software can place the system into a safe state.	
22	Implement safety-critical requirements from the NPR 7150.2.	SWE-023 **[SAFETY CRITICAL ONLY]**	if a project has *safety-critical software*… shall implement the safety-critical software requirements contained NASA-STD-8739.8.	ITA and S&MA
		SWE-219 **[SAFETY CRITICAL ONLY]**	…, if a project has *safety-critical software*, shall ensure that there is 100 percent code test coverage using the MC / DC criterion for all identified safety-critical software components. *Note*: A*ny < 100% coverage shall be reviewed and waived with rationale by the project manager or technical approval authority*	
		SWE-220 **[SAFETY CRITICAL ONLY]**	…, if a project has *safety-critical software*, shall ensure all identified safety-critical software components have a cyclomatic complexity value of 15 or lower. Any exceedance shall be reviewed and waived with rationale by the project manager or technical approval authority	ITA / S&MA
23	Determine which processes, documents, electronic products, activities, and tasks are required.	SWE-036	… shall establish and maintain the software processes, software documentation plans, list of developed electronic products, deliverables, and list of tasks for the software development that are required for the project's software developers, as well as the action required (e.g., approval, review) of the Government upon receipt of each of the deliverables.	Yes
24	Define milestones at which progress will be reviewed and audited..	SWE-037	… shall define and document the milestones at which the software developer(s) progress will be reviewed and audited.	Yes
25	Require supplier to allow NASA to: a. Monitor product integration. b. Review verification activities. c. Review trade studies and source data. d. Audit software development process.	SWE-039	… shall require the software developer(s) to periodically report status and provide insight into software development and test activities; at a minimum, the software developer(s) will be required to allow the project manager and software assurance personnel to: a. Monitor product integration. b. Review the verification activities to ensure adequacy. c. Review trades studies and source data. d. Audit the software development processes and practices. e. Participate in software reviews and technical interchange meetings.	Yes

	e. Participate in software reviews and technical interchange meeting (TIMs).			
	Require the software developers to provide NASA with software products and process tracking information in electronic format, including software development and management metrics. *Note: The number of software defects assessed against mission success risks for proper resolution (vs. total number of software defects) can be used to meet the intent of software quality metric requirements to collect, analyze, and report out to the project per Class I and Class II.*	SWE-040	… shall require the software developer(s) to provide NASA with software products, traceability, software change tracking information and nonconformances, in electronic format, including software development and management metrics.	Yes
26	*[CAP053:023]*	SWE-206	… shall require the software developers and custom software suppliers to provide NASA with electronic access to the models, simulations, and associated data used as inputs for auto-generation of software.	Yes
27	Require the software developer to provide NASA with electronic access to source code in modifiable format, including MOTS software, non-flight software.	SWE-042	… shall require the software developer(s) to provide NASA with electronic access to the source code developed for the project in a modifiable format.	Yes
28	Participate in joint NASA / supplier audits	SWE-045	… shall participate in any joint NASA / developer audits.	Yes

	of the software development process and software CM process.			
29	Require supplier to provide schedule and schedule updates, as required.	SWE-046	... shall require the software developer(s) to provide a software schedule for the project's review and schedule updates, as requested.	Yes
30	Specify re-usability requirements that apply to software development activities to enable future reuse of the software, including models used to generate the software.	SWE-147	… shall specify reusability requirements that apply to its software development activities to enable future reuse of the software, including the models, simulations, and associated data used as inputs for auto-generation of software, for United States Government purposes. [as required]	Yes
31	Plan/support for software assurance activities.	SWE-022	…shall plan and implement software assurance per NASA-STD-8739.8.	Yes
32	Evaluate software for potential reuse and contribute reuse candidates to Agency Software Catalog.	SWE-148	… shall evaluate software for potential reuse by other projects across NASA and contribute reuse candidates to the NASA Internal Sharing and Reuse Software systems. However, if the project manager is a contractor, then a civil servant needs to pre-approve all such software contributions; all software contributions should include, at a minimum, the following information: a. Software Title. b. Software Description. c. The Civil Servant Software Technical Point of Contact for the software product. d. The language or languages used to develop the software. Any third-party code contained therein, and the record of the requisite license or permission received from the third party permitting the Government’s use and any required markings (e.g., required copyright, author, applicable license notices within the software code, and the source of each third-party software component (e.g., software URL & license URL)), if applicable. e. Release notes.	Yes
33	Document provisions for the use of auto-code in the software development process	SWE-146	…shall define the approach to the automatic generation of software source code including: a. Validation and verification of auto-generation tools. b. Configuration management of the auto-generation tools and associated data.	Yes

			c. Description of the limits and the allowable scope for the use of the auto-generated software. d. Verification and validation of auto-generated source code using the same software standards and processes as hand-generated code. e. Monitoring the actual use of auto-generated source code compared to the planned use. f. Policies and procedures for making manual changes to auto-generated source code. g. Configuration management of the input to the auto-generation tool, the output of the auto-generation tool, and modifications made to the output of the auto-generation tools.	
34	Plan for requirements, documentation, rights, support, V&V, and vendor defect reporting of COTS, GOTS, MOTS, or reused software components.	SWE-027	... shall satisfy the following conditions when a COTS, GOTS, MOTS, or reused software component is acquired or used: a. The requirements to be met by the software component are identified. b. The software component includes documentation to fulfill its intended purpose (e.g., usage instructions). c. Proprietary rights, usage rights, ownership, warranty, licensing rights, and transfer rights have been addressed. d. Future support for the software product is planned and adequate for project needs. e. The software component is verified and validated to the same level required to accept a similar developed software component for its intended use. f. The project has a plan to perform periodic assessments of vendor reported defects to ensure the defects do not impact the selected software components.	Yes
35	Software design into Software code	SWE-060	... shall implement the software design into software code.	Yes
36	Perform static analysis.	SWE-135	… shall use static analysis tools to analyze the code during the development and testing phases to detect defects, software security, and coding errors.	Yes
37	Unit Test (For Class I, MC and DC; For Class II, Decision Coverage)	SWE-062	… shall unit test the software code.	Yes
		SWE-186	… shall assure that the unit test results are repeatable.	Yes
38	Software Test	SWE-066	... shall test the software against its requirements	Yes
		SWE-187	… shall place software items under configuration management prior to testing.	Yes
		SWE-189	… shall ensure that the code coverage measurements for the software are selected, implemented, tracked, recorded, and reported. *Note: Percent coverage value is agreed and signed off by the Center's Engineering TA and measured*	Yes
		SWE-190	… shall verify through test the software requirements that trace to a hazardous event, cause, or mitigation technique.	ITA / S&MA

		SWE-191	… shall plan and conduct software regression testing to demonstrate that defects have not been introduced into previously integrated or tested software and have not produced a security vulnerability.	Yes
		SWE-192	… shall verify through test the software requirements that trace to a hazardous event, cause, or mitigation technique.	ITA / S&MA
		SWE-211	… shall test embedded COTS, GOTS, MOTS, OSS, or reused software components to the same level required to accept a custom developed software component for its intended use.	Yes
39	Perform validation on targeted platform or high-fidelity simulation.	SWE-073	... shall validate the software system on the targeted platform or high-fidelity simulation.	Yes
40	Software Cybersecurity / Work with Project Cybersecurity Analyst	SWE-154	… shall identify cybersecurity risks, along with their mitigations, in flight and ground software systems and plan the mitigations for these systems.	ITA / Cybersecurity
		SWE-156	… shall perform a software cybersecurity assessment on the software components per the Agency security policies and the project requirements, including risks posed by the use of COTS, GOTS, MOTS, OSS, or reused software components.	ITA / Cybersecurity
		SWE-157	… shall implement protections for software systems with communications capabilities against unauthorized access per the requirements contained in the NASA-STD-1006, Space System Protection Standard.	ITA / Cybersecurity
		SWE-159	… shall test the software and record test results for the required software cybersecurity mitigation implementations identified from the security vulnerabilities and security weaknesses analysis.	ITA / Cybersecurity
		SWE-185	… shall verify that the software code meets the project's secure coding standard by using the results from static analysis tool(s).	ITA / Cybersecurity
		SWE-207	… shall identify, record, and implement secure coding practices. *Note: For secure coding practices guidance, see nen.nasa.gov/web/coding/links*	ITA / Cybersecurity
		SWE-210	… shall identify software requirements for the collection, reporting, and storage of data relating to the detection of adversarial actions.	ITA / Cybersecurity
41	Software Non-Conformance or Defect Management	SWE-201	… shall track and maintain software non-conformances (including defects in tools and appropriate ground software).	ITA / S&MA
		SWE-202	… 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).	ITA / S&MA
		SWE-203	… shall implement mandatory assessments of reported non-conformances for all COTS, GOTS, MOTS, OSS, and / or reused software components.	ITA / S&MA

42	Write a VDD that includes: a. What the software is supposed to do b. How to install the software c. How to operate the software d. List of software components and version numbers e. List of incorporated and outstanding bug fixes	SWE-063	Shall provide a software version description for each software release.	Yes
		AFOP-7150.2-004* (Flight Software Media Control 1)	Prior to installation on an aircraft, a VDD will be produced containing a form AFRC 80184 and attached a form AFRC 70010 requesting installation.	Yes
43	Prepare software media as required for installation	AFOP-7150.2-004* (Flight Software Media Control 2)	All software media (tape, disk or chip) … identified and physically labeled at the time of production	Yes
44	Prepare the form AFRC 80184	AFOP-7150.2-004* (Flight Software Media Control 3)	…form AFRC 80184 that uniquely identifies (via checksum(s), file size / modification dates, or other verifiable means) the specific software load that should be installed on the aircraft	Yes
		AFOP-7150.2-004* (Flight Software Media Control 4)	Flight software for a specific flight or block of flights … on a form AFRC 80184	Yes
45	Prepare the software installation procedure	AFOP-7150.2-004* (Flight Software Media Control 5)	A procedure … written for flight software installation into the aircraft computer and for verification of correct loading.	Yes

		AFOP-7150.2-004* (Flight Software Media Control 6)	Quality inspection shall verify the correct flight software is loaded for the specified flight according to approved procedures	Yes

Chapter 5: Documentation / Artifacts from SWEHB

This section explains the artifact requirement for each software life-cycle phase. To use this section, first determine the classification level of the software by using the 2-step process described in “Attachment A - Software Classification Worksheet Template” of AFOP-7150.2-004 (or the flowchart associated with Attachment A), then select the column that matches the software classification level. This is a complete list as detailed in NASA-HDBK-2203, NASA Software Engineering Handbook. The SDA within their SDP, or contractor’s approved equivalent SMP, are expected to detail how they plan to meet the intent of the required artifacts. Artifacts can be documents, review presentation material, drawings, source code, and executables. Artifacts can be combined or exist as formal review records. For small projects, documents may be combined to meet the intent of the SWE / deliverable with the Center Software’s TA (or delegate) approval.

Documents marked with an “X*” under Class IV are required for ***FLIGHT*** software only.

Table 5-1. SWEHB Documentation / Artifacts

Source SWEHB	Source NPR (SWE)	Description	C	I/II	D	III	E	IV
		Software Plans from SWEHB						
SMP-SDP	013, 125, 176	Software Management Plan (SMP) – (including NPR 7150.2 compliance matrix)	X	X	X	X	X	X
SDP-SMP	013, 125, 176	Software Development Plan (SDP) from SDAs – (including SMP / NPR 7150.2 compliance matrix)	X	X	X	X	X	X
SCMP	079, 084, 187	Software Configuration Management Plan (SCMP) – in place prior to testing	X	X	X	X		
STP	065, 071	Software Test Plan (STP)	X	X	X	X		
Maintenance	075	Software Maintenance Plan (includes Retirement), Maintenance per classification, Records / Tools Archive (Maint)	X	X	X	X		
SAP	022	Software Assurance Plan (SAP)	X	X	X	X		
Safety	023, 134	Software Safety Plan, if safety-critical software (Safety) -SSP (can be combined with SAP → Software Safety Quality Assurance Plan) Safety Critical (SC)	X(SC)	X(SC)				
		Documents from SWEHB						
Risk	086	Records of Continuous RMP	X	X				
Train	017	Software Training Plan (Train)	X	X				
SRS	050, 051	Software Requirements Specification (SRS)	X	X	X	X	X	X (informal)
IDD	050, 051, 058, 065	Interface Design Description (IDD)	X	X				
SWDD	057	Software Design Description (Architectural Design) - SWDD – architectural	X	X				
SWDD	058	Software Design Description (Detailed Design) - SWDD – detailed	X	X				
SDD	050, 058	Software Data Dictionary (SDD)	X	X				
Test	062, 065, 071	Software Test Procedures (Test)	X	X	X	X		X*
STR	065, 068	Software Test Reports (STR) (including analyses)	X	X	X	X		X*
SUM	058, 077	Software User's Manual (SUM)	X(SC)	X(SC)	X			
Metrics	018, 040	Software Metrics / Tech Perf Metrics (TPM)	X	X	X			

SW VDD	063, 081 084, AFRC	Software Version Description (SW VDD) - including Discrepancy Reports / CR / Problem Reports ((post baseline / release)	X	X	X	X		X*
Inspect	087, 088, 089	Peer Reviews (Inspection)	X	X				
Entrance / Exit	015, 151	Software Cost Estimate	X	X	X	X		
Source SWEHB	Source NPR (SWE)	Description	C	I / II	D	III	E	IV
Entrance / Exit	027, 050,156, 202, 203, 211	Requirements on OTS Software and Reuse	X	X	X			
Entrance / Exit	051, 094	Software (Requirements / Design) Analyses: functional, testable, operable, Failure Modes and Effects Analysis (FMEA), reliability, safety / hazards, life cycle, security	X	X				
Entrance / Exit	020	Preliminary Hazard Analysis / Software Classification / Criticality Test for Safety Critical (PHA)	X	X	X	X	X	X
Entrance / Exit	055, 065, 071	V&V Plan	X	X	X	X		
Entrance / Exit	052	Bi-directional traceability matrix	X	X	X	X		
Entrance / Exit	070, 135, 146	Software Tools	X	X				
Entrance / Exit	033, 039	Record of trade-off criteria & assessment (make / buy decision)	X	X	X			
Entrance / Exit	090, 093, 094	Measurement Analysis Results	X	X				
Entrance / Exit	034	Acceptance Criteria Conditions	X	X	X	X		
		Additional SWE documents						
	016, 018, 046	Software Schedule	X	X	X	X		
	135, 189, 190 219	Code Coverage Test / Analysis Results	MC / DC	MC / DC	PATH	PATH		
	191	Regression Testing	X	X				
	022, 061, 146 157, 185, 195	Coding Standards and Static Analysis Tools Check	X	X	X			
	027, 135, 191 194, 201	Software Defects / Bugs Tracking, (includes unit testing for safety critical code) – including post baseline	X	X	X	X		
	027, 135, 191 194, 201	Software Defects / Bugs Severity Levels & Formal Assessments2, COTS – including post baseline	X	X				

	146. 147, 206	Requirements on Automatic Generation of Software Source Code	X	X				
	068, 159, 176 185, 186, 190	Software Analysis (test results only)	X	X				
	134	Technical Software Safety Features Check	X	X	X	X		
	156	Cybersecurity Assessment	X	X	X	X	X	X
	154, 159	Cybersecurity Analysis	X	X	X	X		
	034	Acceptance Criteria Conditions	X	X	X	X		
	036, 077, 194	Documentation Plan, Electronic Products List, Deliverables, Developers Task Lists, Governmental Approval	X	X	X			
	201, 202, 203	Non-Conformance	X	X				
Source SWEHB	Source NPR (SWE)	Description	C	I / II	D	III	E	IV
	077, 194, 084	Functional Configuration Audit (FCA) (req V&V, version, open items, changes, sign-off, as-built docs / records, CCB process)	X	X	X			
	077, 085	Physical Configuration Audit (PCA) (config items, HRs, config status, safety / security impacts, storage, processing, distribution, release, post support)	X	X	X			
	AFRC	Flight Media Release (FMR AFRC 80184) (load at AFRC)	X	X	X	X		X*

5.1 Summary of Deliverables by Life-Cycle Phase

D – Draft **P** – Preliminary **B** – Baseline **U** – Updated / Updated as required **F** - Final
X(F) – assume complete (final) not explicit in NPRs or NASA Handbook **PPS** – Per Project Schedule

MCR = Mission Concept Review, **SRR** = System Requirements Review, **SWRR** = Software Requirements Review, **MDR** = Mission Definition, Review, **SDR** = System Definition Review, **PDR** = Preliminary Design Review, **CDR** = Critical Design Review, **SIR** = System Integration Review, **I&T** = Integration and Test, **TRR** = Test Readiness Review, **SAR** = System Acceptance Review, **SW Deliv** = Software Delivery, **ORR** = Operational Readiness Review

Table 5-2. Deliverables by Life-Cycle Phase

Product (from NASA-HDBK-2203 Section 7.8 Maturity of Life Cycle Products at Milestone Reviews)	MCR	SRR / SWRR	MDR	SDR	PDR	CDR	I&T SIR	TRR	SW Deliv SAR	ORR
Software Management Plan (SMP)		P	P	B	U	U	U	U	F	
SDA Software Development Plan SDP)		P	P	B	U	U	U	U	F	
Software Schedule	D	P	U	U	B	U	U	U	U	
Software Cost Estimate	D	P	U	U	B	U	U	U	U	
Configuration Management Plan (CMP)		P	P	P	B	U	U	U	U	
Software Test Plans (STP)					P	B	U	U	F	
Software Test Procedures (Tests)						P	U	B	F	
Regression Testing							P	B	F	
Product (from NASA-HDBK-2203 Section 7.8 Maturity of Life	MCR	SRR/ SWRR	MDR	SDR	PDR	CDR	I&T SIR	TRR	SW Deliv SAR	ORR

Cycle Products at Milestone Reviews) continued										
Software Test Reports (STR)									F	
Software Maintenance Plan (Maint)						D	P	P	B	U/F
Software Requirements Specification (SRS)		P	P	P	B	U	U	U	F	
Software Data Dictionary (SDD)		P	P	P	P	B	U	U	F	
Software Design Description (Architectural Design) (SWDD - architectural)					B	U	U	U	F	
Software Design Description (Detailed Design) (SWDD - detailed)					P	B	U	U	F	
Interface Design Description (IDD)					P	B	U	U	F	
Software User's Manual (SUM)										B
Risk Management Plan (RMP)	P	U	U	U	U	U	U	U	U	
Software Metrics/Tech Perf Metrics (TPM) Actual vs Planned					P	U	B	F	F	
Record of trade-off criteria and assessment (make/buy decision)					X(F)	X(F)				
Acceptance Criteria and Conditions					P	B			F	
Software Assurance & Software Safety Products – NASA-HDBK-2203 Section 8.5 (partial - see	MCR	SRR/ SWRR	MDR	SDR	PDR	CDR	I&T SIR	TRR	SW Deliv SAR	ORR

AFOP 7150.2 Rev F2 for full list)										
Software Quality Assurance Plan (SAP)		D	P	B	U	U	U	U	F	
Software Safety Plan (for safety critical)		P			B	U	U	U	F	
Software Analysis Tools, Models, Sims, Validation/ Accreditation		P			B	U	U	U	F	
Coding Standards and Static Analysis Tools Check							F			
Modified Conditions and Decision Code (MC/DC) Coverage Test/Analysis Results (target % set by SW TA, risk assessment if not meet target or uncovered code, cyclomatic complexity < = 15, risk assessment if not <= 15, loaded data, uplinked data, rules, scripts, config data, no dead code)							B	U	F	
Path Code Coverage Test/Analysis Results (target % set by SW TA, risk assessment if not meet target or uncovered code)							B	U	F	
Software Assurance & Software Safety Products – NASA-HDBK-2203 Section 8.5 (partial - see AFOP 7150.2 Rev F2 for full list) continued	MCR	SRR/ SWRR	MDR	SDR	PDR	CDR	I&T SIR	TRR	SW Deliv SAR	ORR

Software Defects / Bugs Tracking							PPS	PPS	PPS	PPS
Software Defects / Bugs Severity Levels and Formal Assessments							PPS	PPS	PPS	PPS
Discrepancy Reports (post baseline / release)										U
Peer Review				Results						
Additional Products	MCR	SRR/ SWRR	MDR	SDR	PDR	CDR	I&T SIR	TRR	SW Deliv SAR	ORR
Requirements on OTS S/W and Reuse		P	P	P	B	U	U	U	F	
Preliminary Hazard Analysis/Software Classification/Criticality Test for Safety Critical (PHA)		P	P	P	B	U	U	U	F	
Measurement Analysis Results		P	P	P	X(F)	X(F)			F	
Software Training Plan								PPS	PPS	PPS
V&V Plan		P			B	U	U	U	F	
Bi-directional traceability matrix		P			B	U	U	U	F	
Additional Products continued	MCR	SRR/ SWRR	MDR	SDR	PDR	CDR	I&T SIR	TRR	SW Deliv SAR	ORR
Software Analyses: functional, testable, operable, FMEA, reliability, safety/hazards, life cycle, security		P			B	U	U	U		

Technical Software Safety Features Check – See SWE-134					P	B	U	U	F	
Cybersecurity Assessment		P	U	U	U	B	U	U	F	
Cybersecurity Analysis					P	B	U	U	F	
Documentation Plan, Electronic Products List, Deliverables, Developers Task Lists, Governmental Approval		P	U	U	B	U	U	U	F	
NPR 7150.2 compliance matrix		P	P	B	U	U	U	U	F	
Requirements on Automatic Generation of Software Source Code					P	B	U	U	F	
Data Flow Diagrams – See SWE-057					P	B	U	U	F	
Baseline Software Build (Configuration Control Board)								B	F	
Functional Configuration Audit (FCA)								F		
Physical Configuration Audit (PCA)									F	
Software Version Description (SW VDD)							P	B	F	
Flight Media Release (load at AFRC)									F	

Chapter 6: NPR 7150.2 Compliance/Comparison

This section describes how the Center requirements align with NPR 7150.2 and, in some places, where the Center requirements exceed those of NPR 7150.2.

6.1 Class C to Class I/II

Class I and II (Safety and Non-Safety Critical) map directly to Class C.

6.2 Class D to Class III

Class III (Mission Critical, Non-Safety Critical) maps directly to Class D with the following exceptions:

RETAIN SWE-187 (Post-Baseline only): … shall place software items under configuration management prior to testing.

DELETE SWE-054: … shall identify, initiate corrective actions, and track until closure inconsistencies among requirements, project plans, and software products.

DELETE SWE-055: … shall perform requirements validation to ensure that the software will perform as intended in the customer environment.

DELETE SWE-061: ... shall select and adhere to software coding methods, standards, and criteria.

DELETE SWE-082: ... shall establish and implement procedures to:
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.

DELETE SWE-083: ... shall prepare and maintain records of the configuration status of software configuration items.

DELETE SWE-084: ... 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.

DELETE SWE-135: … shall use static analysis tools to analyze the code during the development and testing phases to detect defects, software security, and coding errors

DELETE SWE-185: … shall verify that the software code meets the project's secure coding standard by using the results from static analysis tool(s).

DELETE SWE-206: … shall require the software developers and custom software suppliers to provide NASA with electronic access to the models, simulations, and associated data used as inputs for auto-generation of software.

DELETE SWE-207: … shall identify, record, and implement secure coding practices

DELETE SWE-023 (SC): if a project has *safety-critical software*… shall implement the *safety-critical* software requirements contained NASA-STD-8739.8.

DELETE SWE-219 (SC): …, if a project has *safety-critical software*, shall ensure that there is 100 percent code test coverage using the MC / DC criterion for all identified *safety-critical* software components.

DELETE SWE-220 (SC): …, if a project has *safety-critical software*, shall ensure all identified *safety-critical* software components have a cyclomatic complexity value of 15 or lower. Any exceedance shall be reviewed and waived with rationale by the project manager or technical approval authority.

MODIFIED SWE-077: (*External Vendors Only*): ... shall complete and deliver the software product to the customer with appropriate records, including as-built records, to support the operations and maintenance phase of the software's life cycle.

MODIFIED SWE-147 (*As Required by ITA*): … shall specify reusability requirements that apply to its software development activities to enable future reuse of the software, including the models, simulations, and associated data used as inputs for auto-generation of software, for United States Government purposes. [as required]

MODIFIED SWE-151 (*If software cost exceeds $2,000,000*): … shall software cost estimate(s) satisfy the following conditions:

a. Covers the entire software life cycle.
b. Is based on selected project attributes (e.g., assessment of the size, functionality, complexity, criticality, reuse code, modified code, and risk of the software processes and products).
c. Is based on the cost implications of the technology to be used and the required maturation of technology.
d. Incorporates risk and uncertainty, including end state risk and threat assessment for cybersecurity.
e. Includes the cost of the required software assurance support.
f. Includes other direct costs.

6.3 Class E to Class IV

Class IV is a superset of Class E. Class IV contains all of the items within Class E with the following additional items required for ***FLIGHT*** software and one deletion at the request of SQA:

RETAIN SWE-050 (Flight Only; exclude COTS / GOTS / MOTS):

RETAIN SWE-053 (Flight Only; exclude COTS / GOTS / MOTS): ... shall track and manage changes to the software requirements.

RETAIN SWE-063 (Flight and External releases only [exclude COTS]): ... shall provide a software version description for each software release.

RETAIN SWE-065 (Flight and External releases only): … shall establish and maintain:
a. Software test plan(s).
b. Software test procedures(s).
c. Software test report(s)

RETAIN SWE-066 (Flight Only; exclude COTS / GOTS / MOTS): ... shall perform software testing.

RETAIN SWE-071 (Flight and External releases only): …shall update software test plan(s) and software test procedures(s) to be consistent with software requirements.

RETAIN SWE-080 (Flight Only; exclude COTS / GOTS / MOTS): ... shall track and evaluate changes to software products.

RETAIN SWE-081 (Flight Only; exclude COTS): ... shall identify the software configuration items (e.g., software records, code, data, tools, models, scripts) and their versions to be controlled for the project.

DELETE SWE-042: … shall require the software developer(s) to provide NASA with electronic access to the source code developed for the project in a modifiable format.

6.4 Center Airworthiness Requirements on All Center Flight Software (Flight Software Media Control (FSWMC))

The following six requirements are enforced on all software that is to be used for flight or non-flight software:

a. FSWMC1: Prior to installation on an aircraft, a VDD will be produced containing an AFRC 80184 and an attached AFRC 70010 requesting installation.

- Maintains requirement of a VDD.

b. FSWMC2 (EXCLUDE Class IV and COT/GOT in Class III): All software media (tape, disk, or chip) identified and physically labeled at the time of production.

c. FSWMC3: An AFRC 80184 that uniquely identifies (via checksum(s), file size/modification dates, or other verifiable means) the specific software load that should be installed on the aircraft.

d. FSWMC4: Flight software for a specific flight or block of flights on an AFRC 80184.

e. FSWMC5: A procedure written for flight software installation into the aircraft computer and for verification of correct loading.

f. FSWMC6: Quality inspection shall verify the correct flight software is loaded for the specified flight according to approved procedures.

6.5 Documents Changed / Deleted

Deleted Concept of Operations from software. This will be created at the project level early in the creation of the program.

Appendix A, Definitions

Accredit. The official acceptance of a software development tool, model, simulation (including associated data) to use for a specific purpose.

Accreditation. The official recognition or certification that a specific software tool meets defined quality standards and is deemed acceptable for a specific purpose or use.

Airworthiness. The capability of an aircraft [or research/science asset] to be operated within a prescribed flight envelope in a safe manner, per NPR 7900.3, Aircraft Operations Management.

Analysis. The post-processing or interpretation of the individual values, arrays, files of data, or execution of information. It is a careful study of something to learn about its parts, what they do, and how they are related to each other.

Application Software. Software designed to help users perform particular tasks or handle particular types of problems, as distinct from software that controls the computer itself. (ISO 24765:2010)

Assure. When personnel make certain that the specified software engineering, software management, and software assurance activities have been performed by others.

Bi-directional Traceability. Association among two or more logical entities that is discernible in either direction (to/from an entity). (ISO/IEC/IEEE 24765:2010, Systems and Software Engineering - Vocabulary)

Code Coverage. The percentage of the software that has been executed (covered) by the test suite.

Commercial Off-The-Shelf (COTS) Software. The software product is available for purchase and use without the need to conduct development activities. COTS solutions, as opposed to custom-developed solutions, are typically readily available in the commercial marketplace and ready for use as purchased.

Computer. Functional unit that can perform substantial computations including numerous arithmetic operations and logic operations.

Computer Software Configuration Item (CSCI). An aggregation of software that is designated for Configuration Management (CM) and treated as a single entity in the CM process.

Computer System. A system containing one or more computers and associated software. (ISO/IEC/IEEE 24765:2010)

Condition. (1) Measurable qualitative or quantitative attribute that is stipulated for a requirement and that indicates a circumstance or event under which a requirement applies (Systems and software engineering--Systems and software assurance--Part 1: Concepts and vocabulary ISO/IEC/IEEE 15026-1:2019, 3.1.5), (2) Description of a contingency to be considered in the representation of a problem or a reference to other procedures to be considered as part of the condition (Information processing -- Specification of single-hit decision tables, ISO 5806:1984, 3.6), and (3) Boolean expression containing no Boolean operators (Software and systems engineering -- Software testing -- Part 4: Test techniques, ISO/IEC/IEEE 29119-4:2015, 4.6).

Control Flow. The sequence in which operations are performed during the execution of a computer program. (ISO/IEC/IEEE 24765:2010)

Coverage Analysis. The process of determining the degree to which a proposed software verification process activity satisfies its objective. (RTCA DO-178C)

Cybersecurity. The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability.

Cyclomatic Complexity. Cyclomatic complexity is a software metric used to indicate the complexity of a program. It is a quantitative measure of the number of linearly independent paths through a function’s source code.

Data. Information for computer processing (e.g., numbers, text, images, and sounds in a form that is suitable for storage in or processing by a computer).

Data Flow. The sequence in which data transfer, use, and transformation are performed during the execution of a computer program. (ISO/IEC/IEEE 24765:2010)

Deactivated Code. Executable object code (or data) that is traceable to a requirement and, by design, is either not intended to be executed (code) or used (data) or is only executed (code) or used (data) in certain configurations of the target computer environment. (RTCA DO-178C)

Dead Code. Executable object code (or data) that exists as a result of a software development error but cannot be executed (code) or used (data) in any operational configuration of the target computer environment. It is not traceable to a system or software requirement. (RTCA DO-178C)

Decision Coverage. Every point of entry and exit in a program has been invoked at least once and every decision in the program has taken on all possible outcomes at least once. (RTCA DO-178C)

Defect. Any occurrence in a software product that is determined to be incomplete or incorrect relative to the software requirements, expectations for the software, and/or program standards. (Source: NASA-STD-8739.9)

Deviation. A documented authorization releasing a program or project from meeting a requirement before the requirement is put under configuration control at the level the requirement will be implemented.

DO (not an acronym). RTCA document identification.

Embedded Computer System. A computer system that is part of a larger system and performs some of the requirements of that system. (ISO/IEC/IEEE 24765:2010)

Embedded Software. Software that is part of a larger system and performs some of the requirements of that system. (ISO/IEC/IEEE 24765:2010)

Ensure. To secure or guarantee, to make sure or certain.

Establish and Maintain. Formulation, documentation, use/deployment, and current maintenance of the object (usually a document, requirement, process, or policy) by the responsible project, organization, or individual.

Existing Software. Software that is already developed and available; is usable either as is or with modifications; and that is provided by the supplier, acquirer, or a third party. (ISO/IEC/IEEE 24765:2010)

Facility. Used in this directive to show research, development, test, or simulation facilities representing a significant NASA investment (facilities with a Current Replace Value (CRV) equal to or greater than $50,000,000), which contains software that supports programs and projects managed under NPR 7120.5, NPR 7120.7, or NPR 7120.8, NASA Research and Technology Program and Project Management Requirements, and that have a Mission Dependency Index value equal to or greater than 70.

Facility Safety. Safety efforts targeted at industrial activity associated with the access to and operation of all facilities, including special support capabilities that are resident within these facilities. (AFPD-8700.1-001)

Failure. The behavior of the software or system component when a fault is encountered, producing an incorrect or undesired effect of a specified severity. (Source: NASA-STD-8739.9)

Fault. The manifestation of an error in software that may cause a failure. (Source: NASA-STD-8719.24 Annex)

Flight Software. Software that directly modifies or monitors vehicle operation, whether the software is installed in a system on-board an aircraft or installed in a ground-based system that modifies/monitors aircraft operation. (This directive)

Firmware. Combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device. (ISO/IEC/IEEE 24765:2010 for programmable firmware see Programmable Logic Device)

Freeware. Software that is proprietary and that is available for use at no monetary cost. In other words, freeware may be used without payment but may usually not be modified, re-distributed, or reverse-engineered without the author's permission.

Glueware. Software created to connect the OTS software / reused software with the rest of the system. It may take the form of "adapters" that modify interfaces or add missing functionality, "firewalls" that isolate the OTS, or "wrappers" that check inputs and outputs to the OTS software and may modify to prevent failures.

Government Off-The-Shelf Software (GOTS). Software supplied by the government for reuse in another project. This refers to Government-created software, usually from another project. The software was not created by the current developers (see software reuse). Usually, source code is included and documentation, including test and analysis results, is available (e.g., the Government is responsible for the GOTS software to be incorporated into another systems). (ISO/IEC/IEEE 24765:2010)

Ground Safety. Safety efforts targeted at activity not included within the definition of flight safety. (AFPD-8700.1-001)

Ground Software. Software that could indirectly impact flight or test operations. This includes software supporting simulation, control room, data processing, or V&V test operations. (This directive)

Highly Specialized Information Technology (IT). Highly Specialized IT is a part of, internal to, or embedded in a mission platform. The platform's function (e.g., avionics, guidance, navigation, flight controls, simulation, radar, etc.) is enabled by IT, but not driven by IT itself (e.g., computer hardware and software to automate internal functions of a spacecraft or spacecraft support system such as spacecraft control and status, sensor signal and data processing, and operational tasking). Highly Specialized IT acquisitions may include full development (where the IT is a primary issue) to modification of existing systems (information architecture is firm and demonstrated in an operational environment) where IT is not an issue. Real time is often critical -- and few opportunities exist to use COTS or GOTS beyond microprocessors and operating systems because these systems are largely unprecedented or largely unique applications. Certain IT considered Mission Critical because the loss of which would cause the stoppage of mission operations supporting real-time on-orbit mission operations is identified as "Highly Specialized" by the Directorate Associate

Administrator. Highly Specialized IT is largely custom, as opposed to COTS or commodity IT systems or applications, and includes coding/applications that are integral parts of the research or science requirements (e.g., Shuttle Avionics Upgrade. Common Engineering IT tools such as Product Lifecycle Management systems, Computer-Aided Design systems, and collaborative engineering systems and environments are not Highly Specialized IT. Representative examples of Highly Specialized IT include Avionics software, real-time control systems, onboard processors, Deep Space Network, spacecraft instrumentation software, wind tunnel control system, human physiology monitoring systems, ground support environment, experiment simulators, Mission Control Center, and Launch cameras. (Source: NPR 2800.1, Managing Information Technology)

Independent Verification and Validation (IV&V). V&V performed by an organization that is technically, managerially, and financially independent of the development organization. (Source: ISO/IEC/IEEE 24765:2010) The NASA requirements for IV&V are defined in the NASA-STD-8739.8**.**

Information Technology. Any equipment or interconnected system or subsystem of equipment that is used in the automatic acquisition, storage, manipulation, management, movement, control, display, switching, interchange, transmission, or reception of data or information by an executive agency. IT also includes computers, ancillary equipment (including imaging peripherals, input, output, and storage devices necessary for security and surveillance), peripheral equipment designed to be controlled by the central processing unit of a computer; software; firmware; and similar procedures, services (including support services), and related resources, but does not include any equipment acquired by a Federal contractor incidental to a Federal contract. (Source: NPR 7120.7, NASA Information Technology and Institutional Program and Project Management Requirements)

Insight. An element of Government surveillance that monitors contractor compliance using Government-identified metrics and contracted milestones. Insight is a continuum that can range from low intensity such as reviewing quarterly reports to high intensity such as performing surveys and reviews. (Source: NPR 7123.1, NASA Systems Engineering Processes and Requirements)

Interface. A shared boundary between two functional units, defined by various characteristics pertaining to the functions, physical signal exchanges, and other characteristics. (ISO/IEC/IEEE 24765:2010)

Interpreter. A computer program that translates and executes each statement or construct of a computer program before translating and executing the next. (ISO/IEC/IEEE 24765:2010)

Legacy and Heritage Software. Software products (architecture, code, requirements) written specifically for one project and then, without prior planning during its initial development, found to be useful for other projects. See software reuse.

Media. Devices or materials that act as a means of transferring or storing software. (RTCA DO-178C)

Mission Critical. Item or function that should retain its operational capability to assure no mission failure (i.e., for mission success - meeting all mission objectives and requirements for performance and safety). (Source: NPR 8715.3)

Model. A description or representation of a system, entity, phenomena, or process. (Source: NASA-STD-7009, Standard for Models and Simulations) Only for the purpose of this directive, the term "model" refers to only those models that are implemented in software.

Modified Off-The-Shelf Software (MOTS). Software product that is already developed and available, usable either ’as is’ or with modification, and provided by the supplier, acquirer, or a third party. (ISO/IEC/IEEE 24765:2010)

When COTS or legacy and heritage software is reused, or heritage software is changed, the product is considered “modified.” The changes can include all or part of the software products and may involve additions, deletions, and specific alterations. An argument can be made that any alterations to the code and design of an off-the-shelf (OTS) software component constitute “modification,” but the common usage allows for some percentage (less than 5% of the code changes) of change before the OTS software is declared to be MOTS software. MOTS may include the changes to the application shell or glueware to add or protect against certain features and not to the OTS software system code directly. When less than 30% of the existing code changes, the product can be considered “modified.” If more than 30% of the code changes or if the new code is added, the software should be considered a new software development.

Modified Condition / Decision Coverage (MC / DC).
MC / DC is a code coverage criterion used in software testing. MC/DC requires all of the below during testing: Each condition in a decision is shown to independently affect the outcome of the decision. Independence of a condition is shown by proving that only one condition changes at a time.

Every point of entry and exit in a program has been invoked at least once, every condition in a decision in the program has taken all possible outcomes at least once, every decision in the program has taken all possible outcomes at least once, and each condition in a decision has been shown to independently affect that decision’s outcome. A condition is shown to independently affect a decision’s outcome by varying just that condition while holding fixed all other possible conditions or varying just that condition

while holding fixed all other possible conditions that could affect the outcome. (RTCA DO-178C)

Off-The-Shelf (OTS) Software. Software not developed in-house or by a contractor for the specific project now underway. The software is developed for a purpose different from the current project. Used in practice as an umbrella for COTS, GOTS, MOTS, OSS, freeware, shareware, trial software, demonstration software, legacy software, heritage software, and reuse software. (NPR 7150.2)

Open-Source Software (OSS). Software where its human-readable source code is made broadly available without cost under an OSS license, which provides conditions on use, reuse, modification/improvement, and redistribution; and often where the software development, management, and planning is done publicly, or easily observable by an individual or organization not previously connected with its open-source project. (NPR 7150.2)

Operational Software. Software that has been accepted and deployed, has been delivered to its customer, or is deployed in its intended environment. (NPR 7150.2)

Partitioning. A technique for providing isolation between software components to contain and / or isolate faults. (RTCA DO-178C)

Primary Mission Objectives. Outcomes expected to be accomplished, which are closely associated with the reason the mission was proposed, funded, developed, and operated (e.g., objectives related to top-level requirements or their flow down).

Process Asset Library. A collection of process asset holdings that may be used by an organization or project. (Source: Capability Maturity Model Integration (CMMI®) for Systems Engineering/Software Engineering/Integrated Product and Process Development Supplier Sourcing)

Program. A strategic investment by a Mission Directorate or Mission Support Office that has a defined architecture and/or technical approach, requirements, funding level, and a management structure that initiates and directs one or more projects. A program defines a strategic direction that the Agency has identified as critical.

Programmable Logic Device. A semiconductor device based on a matrix of configurable logic blocks connected via a configurable interconnect. The circuitry (combinational/sequential logic, memory/storage, input/output) in a PLD is configured to meet design requirements for a desired application after device manufacturing. (NPR 7150.2).

Project. A specific investment having defined goals, objectives, requirements, life-cycle cost, a beginning, and an end. A project yields new or revised products or services that directly address NASA's strategic needs. They may be performed wholly in-house; by

Government, industry, academia partnerships; or through contracts with private industry.

Project Manager. A generic term that represents the position in charge of the project. A project manager could be designated as a project lead, project principal investigator, project scientist, research director, project executive, or some other term, as defined in the project's governing document. A project manager is responsible for the formulation and implementation of the R&T project, per the governing document in coordination with the program manager. (NPR 7120.5 defines the roles and responsibilities for this position).

Requirements Volatility. The total number of requirements compared to requirement changes over time. It may include additions, changes, and reduction of requirements.

Risk Management. An organized, systematic decision-making process that efficiently identifies, analyzes, plans, tracks, controls, communicates, and documents risk to increase the likelihood of achieving program/project goals. (Source: NPR 8715.3)

Safety Critical. A term describing any condition, event, operation, process, equipment, or system that could cause or lead to severe injury, major damage, or mission failure if performed or built improperly, or allowed to remain uncorrected. (Source NPR 8715.3)

Scripts. A sequence of automated computer commands embedded in a program that tells the program to execute a specific procedure (e.g., files with monitoring, logic, or commands used by software to automate a process or procedure).

Simulation. The imitation of the behavioral characteristics of a system, entity, phenomena, or process. (Source: NASA-STD-7009) Only for the purpose of this document, the term "simulation" refers to only those simulations that are implemented in software.

Shareware. Software that is available free of charge and often distributed informally for evaluation, after which a fee may be requested for continued use.

Software. In this directive, "software" is defined as (1) computer programs, procedures, and associated documentation and data pertaining to the operation of a computer system (IEEE 828-2012, 2.1), (2) all or a part of the programs, procedures, rules, and associated documentation of an information processing system (ISO/IEC 19770-5:2015, Information Technology, 3.34), (3) program or set of programs used to run a computer (ISO/IEC 26514:2008, Systems and software engineering–requirements for designers and developers of user documentation, 4.46) (4) all or part of the programs which process or support the processing of digital information (ISO/IEC 19770-1:2017, Information Technology – IT asset management – Part 1: IT asset management systems--Requirements, 3.49), and (5) part of a product that is the computer program or the set of computer programs (ISO/IEC/IEEE 26513:2017, Systems and software

engineering–requirements for testers and reviewers of information for users, 3.34). This definition applies to software developed by NASA, software developed for NASA, software maintained by or for NASA, COTS, GOTS, MOTS, OSS, reused software components, auto-generated code, embedded software, the software executed on processors embedded in programmable logic devices (see NASA-HDBK-4008), legacy, heritage, applications, freeware, shareware, trial or demonstration software, and open-source software components.

Software Architecture. The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the properties of those components, and the relationships between them. The term also refers to documentation of a system's software architecture. Documenting software architecture facilitates communication between stakeholders, documents early decisions about high-level design, and allows reuse of design components and patterns between projects.

Software Assurance. The planned and systematic set of activities that ensure that software life cycle processes and products conform to requirements, standards, and procedures. For NASA, this includes the disciplines of Software Quality (functions of Software Quality Engineering, Software Quality Assurance, and Software Quality Control), Software Safety, Software Reliability, Mission Software Cybersecurity Assurance, Software Verification and Validation, and IV&V.

Software Engineering. The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software (i.e., the application of engineering to software). (Source: ISO/IEC/IEEE 24765)

Software Item. Source code, object code, control code, control data, or a collection of these items.

Software Peer Review / Inspection. A visual examination of a software product to detect and identify software anomalies, including errors and deviations from standards and specifications. (Source: IEEE 1028, Standard for Software Reviews and Audits) (Refer to NASA-STD-8739.9 for guidelines for software peer reviews or inspections.)

Software Reuse. A software product developed for one use but having other uses or one developed specifically to be usable on multiple projects or in multiple roles on one project. Examples include, but are not limited to, COTS products, acquirer-furnished software products, software products in reuse libraries, and pre-existing developer software products. (NPR 7150.2)

Software Suppliers. An organization or individual that enters into an agreement with the acquirer for the supply of a software product or service, individual or organization that enters into a contract with the acquirer for the supply of a software system, software product, or software service under the terms of the contract or an organization or part of

an organization or individual that enters into an agreement with the application management organization for the supply of a software product or software service. Software Suppliers includes NASA in-house software development.

Software Tool(s). Software products developed for one use but having other uses, or one developed specifically to be usable on multiple projects or in multiple roles on one project. (ISO/IEC/IEEE 24765:2010)

Software Validation. Confirmation that the product, as provided (or as it will be provided), fulfills its intended use. In other words, validation ensures that "you built the right thing." (Source: IEEE 1012, Standard for Software Verification and Validation)

Software Verification. Confirmation that work products properly reflect the requirements specified for them. In other words, verification ensures that "you built it right." (Source: IEEE 1012, ISO/IEC/IEEE 24765:2010)

Statement Coverage. Every statement in the program has been invoked at least once. (RTCA DO-178C)

Static Analysis. The process of evaluating a system or component unit testing based on its form, structure, content, or documentation. (ISO/IEC/IEEE 24765:2010)

Structural Coverage Analysis. An evaluation of the code structure, interfaces, exercised during requirements-based testing. (RTCA DO-178C)

Subsystem. A secondary or subordinate system within a larger system. (ISO/IEC/IEEE 24765:2010)

Support Software. Software that aids in the development or maintenance of other software. (ISO/IEC/IEEE 24765:2010)

System. The combination of elements that function together to produce the capability required to meet a need. The elements include hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose. (NPR 7123.1)

System Software. Software designed to facilitate the operation and maintenance of a computer system and associated programs. (ISO/IEC/IEEE 24765:2010)

Tailoring. The process used to adjust a prescribed requirement to accommodate the needs of a specific task or activity (e.g., program or project). Tailoring may result in changes, subtractions, or additions to a typical implementation of the requirement. (NPR 7150.2)

Uncertainty. The estimated amount or percentage by which an observed or calculated value may differ from the true value; a broad and general term used to describe an

imperfect state of knowledge or a variability resulting from a variety of factors including, but not limited to, lack of knowledge, applicability of information, physical variation, randomness or stochastic behavior, indeterminacy, judgment, and approximation. (NPR 8000.4, Agency Risk Management Procedural Requirements)

Unit Testing. (1) Testing of individual routines and modules by the developer or an independent tester (ISO/IEC/IEEE 24765). (2) A test of individual programs or modules in order to ensure that there are no analysis or programming errors (ISO/IEC/IEEE 2382-20, Information Technology – Vocabulary – Part 20: System Development). (3) Test of individual hardware or software units or groups of related units. (ISO/IEC/IEEE 24765)

Validation. (1) Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled (ISO/IEC 25000:2014 Systems and software Engineering--Systems and software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE, 4.41) (ISO/IEC/IEEE 12207:2017 Systems and software engineering--Software life cycle processes, 3.1.71) (ISO/IEC/IEEE 15288:2015 Systems and software engineering--System life cycle processes, 4.1.53) (ISO/IEC TS 24748-1:2016 Systems and software engineering--Life cycle management--Part 1: Guide for life cycle management, 2.61), (2) process of providing evidence that the system, software, or hardware and its associated products satisfy requirements allocated to it at the end of each life cycle activity, solve the right problem (e.g., correctly model physical laws, implement business rules, and use the proper system assumptions), and satisfy intended use and user needs (IEEE 1012, Standard for Software Verification and Validation, 3.1.35-2016), (3) the assurance that a product, service, or system meets the needs of the customer and other identified stakeholders. It often involves acceptance and suitability with external customers. (A Guide to the Project Management Body of Knowledge (PMBOK(R) Guide) -- Fifth Edition), and (4) process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements (IEEE 1012, Standard for Software Verification and Validation, 3.1-2016)

> *Note*: *Validation in a system life cycle context is the set of activities ensuring and gaining confidence that a system is able to accomplish its intended use, goals, and objectives. The right system has been built. Validation demonstrates that the system can be used by the users for their specific tasks. "Validated" is used to designate the corresponding status. [ISO 9000] Multiple validations can be carried out if there are different intended uses.*

Verification. Proof of compliance with specifications. Verification may be determined by test, analysis, demonstration, and inspection. (NPR 7123.1)

Waiver. A documented authorization releasing a program or project from meeting a requirement after the requirement is put under configuration control at the level the requirement will be implemented.

Wrapper. See glueware definition.

Appendix B, Abbreviations and Acronyms

AFG	Armstrong Flight Research Center Guide
AFOP	Armstrong Flight Research Center Operational Procedures
AFPD	Armstrong Flight Research Center Policy Directives
AFPR	Armstrong Flight Research Center Procedural Requirements
AFRC	Armstrong Flight Research Center
CCB	Configuration Control Board
CE	Chief Engineer
CIO	Chief Information Officer
CLT	Capability Leadership Team
CM	Configuration Management
CMMI®	Capability Maturity Model Integration
CMP	Configuration Management Plan
CO	Contracting Officer
CONFIG	Configuration
COR	Contracting Officer Representative
COTS	Commercial-Off-The-Shelf
CR	Change Requests
CSCI	Computer Software Configuration Item
DELIV	Delivery
DOCS	Documents
FAM	Flight Assurance Matrix
FAR	Federal Acquisition Regulation

FCA	Function Configuration Audit
FMEA	Failure Modes and Effects Analysis
FSL	Flight Systems Lead
FSWMC	Flight SoftWare Media Control
GOTS	Government-Off-The-Shelf
HDBK	Handbook
IDD	Interface Design Description
IEC	IEC Electronics (company name)
IEEE	Institute of Electrical and Electronic Engineers
ISO	International Standards Organization
IT	Information Technology
ITA	Independent Technical Authority
IV&V	Independent Verification and Validation
KSLOC	Kilo/Thousand Source Lines of Code
MC/DC	Modified Condition / Decision Coverage
MOTS	Modified-Off-The-Shelf
NASA	National Aeronautics and Space Administration
NFS	NASA FAR Supplement
NPR	NASA Procedural Requirements
NSEH	NASA Software Engineering Handbook
OSS	Open-Source Software
OTS	Off-The-Shelf

PCA	Physical Configuration Audit
PCE	Project Chief Engineer
PHA	Preliminary Hazard Analysis
PLD	Programmable Logic Device
PM	Project Manager
RMP	Risk Management Plan
RTCA	Radio Technical Commission for Aeronautics
S&MA	Safety and Mission Assurance
SAP	Software Assurance Plan
SC	Safety Critical
SCMP	Software Configuration Management Plan
SDA	Software Development Agent
SDD	Software Data Dictionary
SDP	Software Development Plan
SE	Software Engineering
SHA	System Hazard Analysis
SM	Software Manager
SMP	Software Management Plan
SOW	Statements of Work
SRS	Software Requirements Specification
SQA	software quality assurance
SSP	System Safety Plan
SSWG	System Safety Working Group

STD	Standard
STP	Software Test Plan
STR	Software Test Reports
SUM	Software User’s Manual
SWDD	software design document
SWE	Software Engineering
SWEHB	Software Engineering Handbook
TA	Technical Authority
TIM	Technical Interchange Meeting
TPM	Technical Performance Metrics
URL	Uniform Resource Locator
U.S.	United States
V&V	Verification and validation
VDD	Version Description Document
WBS	Work Breakdown Structure

Appendix C, Verification Matrix

See Chapter 4 of this document.

Appendix D, Requirements Mapping Matrix

SWE numbers marked with an X and an asterisk (X*) under Class IV are required for ***FLIGHT*** software only. Those marked as (SC) are for Safety-Critical only.

SWE (NPR 7150.2)	C	I/II	D	III	E	IV
SWE-013	X	X	X	X	X	X
SWE-015	X	X	X	X		
SWE-016	X	X	X	X		
SWE-017	X	X				
SWE-018	X	X				
SWE-020	X	X	X	X	X	X
SWE-022	X	X	X	X	X	X
SWE-023	X	X (SC)	X (SC)			
SWE-024	X	X	X	X		
SWE-027	X	X	X	X		
SWE-033	X	X	X	X	X	X
SWE-034	X	X	X	X		
SWE-036	X	X	X	X		
SWE-037	X	X	X	X		
SWE-039	X	X	X	X		
SWE-040	X	X	X	X		
SWE-042	X	X	X	X	X	
SWE-045	X	X				
SWE-046	X	X	X	X		
SWE-050	X	X	X	X		X* – Flight Only; exclude COTS / GOTS / MOTS
SWE-051	X	X				
SWE-052	X	X	X	X		
SWE-053	X	X	X	X		X* – Flight Only; exclude COTS / GOTS / MOTS
SWE-054	X	X	X			
SWE-055	X	X	X			
SWE-057	X	X				
SWE-058	X	X				
SWE-060	X	X				
SWE-061	X	X	X			
SWE-062	X	X	X	X		
SWE-063	X	X	X	X		X* - Flight and External releases only [exclude COTS]
SWE-065	X	X	X	X		X* - Flight and External releases only

SWE (NPR 7150.2)	C	I/II	D	III	E	IV
SWE-066	X	X	X	X		X* – Flight Only; exclude COTS / GOTS / MOTS
SWE-068	X	X	X	X		
SWE-070	X	X				
SWE-071	X	X	X	X		X* - Flight and External releases only
SWE-073	X	X				
SWE-075	X	X	X	X		
SWE-077	X	X	X	X (External Vendors Only]		
SWE-079	X	X	X	X		
SWE-080	X	X	X	X		X* – Flight Only; exclude COTS / GOTS / MOTS
SWE-081	X	X	X	X		X* – Flight Only; exclude COTS
SWE-082	X	X	X			
SWE-083	X	X	X			
SWE-084	X	X	X			
SWE-085	X	X	X	X		
SWE-086	X	X				
SWE-087	X	X				
SWE-088	X	X				
SWE-089	X	X				
SWE-090	X	X				
SWE-093	X	X				
SWE-094	X	X				
SWE-121	X	X	X	X	X	X
SWE-125	X	X	X	X	X	X
SWE-134	X	X	X	X		
SWE-135	X	X	X			
SWE-136	X	X				
SWE-139	X	X	X	X	X	X
SWE-143	X	X - if the development is a payload				
SWE-146	X	X				
SWE-147	X	X	X	X – as required by ITA		
SWE-148	X	X	X	X	X	X
SWE-151	X	If software cost >$2,000,000	X	If software cost >$2,000,000		
SWE-154	X	X	X	X		
SWE-156	X	X	X	X	X	X

SWE (NPR 7150.2)	C	I/II	D	III	E	IV
SWE-157	X	X	X	X		
SWE-159	X	X	X	X		
SWE-174	X	X	X	X		
SWE-176	X	X	X	X	X	X
SWE-184	X	X				
SWE-185	X	X	X			
SWE-186	X	X	X	X		
SWE-187	X	X		X (post baseline only)		
SWE-189	X	X	X	X		
SWE-190	X	X				
SWE-191	X	X				
SWE-192	X	X	X	X		
SWE-194	X	X	X	X		
SWE-195	X	X	X	X		
SWE-196	X	X	X	X		
SWE-199	X	X				
SWE-201	X	X	X	X		
SWE-202	X	X				
SWE-203	X	X				
SWE-205	X	X	X	X	X	X
SWE-206	X	X	X			
SWE-207	X	X	X			
SWE-210	X	X				
SWE-211	X	X				
SWE-219	X	X (SC)	X (SC)			
SWE-220	X	X (SC)	X (SC)			
AFPR-7150.2-001 (FSWMC 1)		X		X		X*
AFPR-7150.2-001 (FSWMC 2)		X		X (exclude COTS/GOTS)		
AFPR-7150.2-001 (FSWMC 3)		X		X		X*
AFPR-7150.2-001 (FSWMC 4)		X		X		X*
AFPR.7150.2-001 (FSWMC 5)		X		X		X*
AFPR-7150.2-001 (FSWMC 6)		X		X		X*

Document History Log

This page is for informational purposes and does not have to be retained with the document.

Baseline, 06-07-10

Baseline-1, Admin Change, 08-17-10

- Formatting changed to comply with Agency standards.

Revision A, 09-03-14

- Answers findings 439-01 and 439-06 OCE
- Updated to address changes associated with NPR 7150.2
  - Minor changes to verbiage in 7150.2 are denoted by an “A” affixed to the DPR requirement number
  - Removed R.0390, R.0400, R.0410, R.0420, R.0430, R.0440
  - Documents the center’s approach to meeting the intent of the CMMI requirements (R.0590)
  - Requires that center software metrics be established and reported
  - Requires a software safety plan for safety-critical software (R.1812, R.1813)
  - Requires an IV&V Project Execution Plan for those projects selected for IV&V (R.0531)
  - Changes the software classification process (R.0531, R.0532)
  - Requires certain features in safety-critical software (R.1271)
  - Requires the use of static analysis tools (R.1001)
  - Requires validation and accreditation of tools (R.2771)
  - Requires peer review of plans (R.1121)
  - Implements compliance matrix (R.0221, R.0222)
- Updated to address changes associated with NASA-STD-8719.13C
  - Implements software safety criticality assessment and software safety litmus test
  - Requires trace of relationships between software safety requirements and software-related system hazards, controls, conditions and events
  - Requires waiver of applicable requirements that are not met
  - Requires safety criticality determination of tools and COTS software
  - Requires that S&MA sit on project decision bodies, review discrepancy reports and approve changes to software critical software
  - Requires that software safety organization participate in evaluation of certification process
  - Requires a software safety plan
  - Requires projects to provide proper resources for software safety
  - Removes requirements related to establishing an official certification process for safety-critical software
  - Refines the definition of safety-critical software
  - Removed duplication between NASA-STD-8719.13 and NPR 7150.2
- Updated to address changes associated with RTCA DO-178C

  - Added and refined requirements to ensure correct relationship is maintained between software components across an interface boundary. (R.0926, R.1002, R.1004)
  - Refined dead code requirements to allow analysis to show that dead code can remain as long as it can be removed during the compile/link process (R.1006)
- Updated descriptions associated with software classification III-S
- Revised Systems Engineering and software lifecycle information.
- Added Appendix F Compliance/Reference Information.

Admin Change, A-1, 12-30-14
- Added Section F.5, F.5.1, F.5.2 and F.5.3
- Answers OCE Finding 439-03

Revision B, 05-09-19
- Formatted to current template
- Updated sections P.2, P.4, Chapters 1, 2, 4, 5, and Appendices A, B, C, D
- Updated document references
- Renumbered paragraphs in Chapter 3
- Added Chapter 6
- Changes address audit findings 553-023 and 553-027

Revision B-1, 09-05-19
- Updated cancelled reference AFOP-8739.8-001, Software Assurance Audit and Corrective Action Procedure, to AFG- 8739.8-002, Software Assurance Audit and Corrective Action Handbook

Revision B-2, 12-05-19
- Removed cancelled references AFOP-7900.3-022, Tech brief (T/B) & Mini Tech Brief (Mini T/B), from section P.4 and renumbered

Admin Change B-3, 05-04-21
- Added page break between chapters.
- Updated use of “must” to “shall”.
- Added Appendix E, Requirement Verification Matrix.

Admin Change B-4, 07-03-23
- Updated obsolete reference AFOP-8715.3-005to AFG-8715.3-018.

Admin Change B-5, 07-19-24
- Expiration date extended from 7/23/24 to 1/23/25.

Admin Change B-6, 01-23-25
Expiration date extended from 1/23/25 to 7/23/25.

Admin Change B-7, 07-23-25

Expiration date extended from 7/23/25 to 1/23/26.

Revision C, 07-30-25

- Updated to incorporate NPR 7150.2 Rev D
  - SW requirements per Classification
  - Reports
  - Frequency/Schedule of reports
  - Definitions
  - Requirements Mapping Matrix
- Included previous change requests from Tech Reviewers
- Moved all referenced and informational documents to P.4.