



**ARMSTRONG
PROCEDURAL
REQUIREMENT (DPR)**

**Directive:
Effective Date:
Expiration Date:**

**DPR-7150.2-001A-1
September 1, 2014
September 1, 2019**

This document is uncontrolled when printed.
Before use, check the Master List to verify that this is the current version.
Compliance is mandatory.

SUBJECT:

Armstrong Software Engineering Requirements

RESPONSIBLE OFFICE:

X / Office of the Chief Engineer

CONTENTS

PREFACE 3

P.1 Purpose 3

P.2 Applicability 3

P.3 Authority 4

P.4 Applicable Documents 4

P.5 Measurement/Verification 5

P.6 Cancellation 5

CHAPTER 1: INTRODUCTION 6

CHAPTER 2: RESPONSIBILITIES 9

CHAPTER 3. SOFTWARE CLASSIFICATION 14

CHAPTER 4. SOFTWARE ENGINEERING REQUIREMENTS 21

CHAPTER 5. COMPARISON TO NPR 7150.2 36

Appendix A: Definitions 43

Appendix B: Acronyms 47

Appendix C: Reference Documents 48

Appendix D: Designated Governing Authority Allocations 49

Appendix E: Requirements Mapping Matrix 55

Appendix F: Compliance/Reference Information 66

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

THIS PAGE INTENTIONALLY LEFT BLANK

PREFACE

P.1 Purpose

a. This procedural requirements (PR) document brings the Armstrong Flight Research Center (AFRC) (herein after referred to as the Center) into compliance with the National Aeronautics and Space Administration (NASA) Policy Directive (NPD) 2820.4 by capturing the requirements in NASA Procedural Requirements (NPR) 7150.2 and NASA Standard NASA-STD-8719.13. In doing so, this document 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 lifecycle, but instead provides a single set of requirements for center software engineering activities. This will allow organizations at the Center that purchase or develop software the freedom to develop processes tailored to their own needs.

b. In addition to the above, this document modifies the software classification approach from that defined in NPR 7150.2 to a hazard/risk based system. This approach used in this DPR 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 and Centerwide procedure [DCP-S-002](#), Hazard Management Procedure.

c. The requirements and software classification methodology found in this DPR provide the hooks needed to extend the Center classification system to include business and information technology (IT) infrastructure software. This hazard/risk based software classification approach may not make logical sense for all IT based software/services defined in [DPD-2800.2-001](#), Attachment A. If this is the case, IT-based software application developers may choose to directly apply the software classifications and associated requirements found in NPR 7150.2.

P.2 Applicability

a. This DPR is applicable to Center and other NASA employees visiting, detailed, or assigned to the Center on a temporary basis. This language also applies to contractors, grant recipients, or parties to agreements only to the extent specified or referenced in the appropriate contracts, grants, or agreements.

b. The requirements of this DPR cover software created or acquired by or for NASA, including commercial-off-the-shelf software (COTS), government-off-the-shelf software (GOTS), modified-off-the-shelf software (MOTS), open source, reuse, legacy, and heritage software. Requirements in this DPR apply to all of the Agency's product lines containing software systems and subsystems. The applicability of requirements in this DPR to specific systems and subsystems within Agency product lines, programs, and

Before use, check the Master List to verify that this is the current version.

This document may be distributed outside of the Center.

projects is determined through the use of the software classes defined in Chapter 2, in conjunction with the Requirements Mapping Matrix in Appendix E. It is not uncommon for a project to contain multiple systems and subsystems having different software classes. Through the use of the Requirements Mapping Matrix, the number of applicable requirements and their associated rigor are scaled back for less critical software classes.

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

P.3 Authority

- a. NPD 2800.1, Managing Information Technology
- b. NPD 7120.4, NASA Engineering and Program/Project Management Policy
- c. NPR 7150.2, NASA Software Engineering Requirements
- d. NPR 8621.1, NASA Procedural Requirements for Mishap and Close Call Reporting Investigating, and Record keeping
- e. NPR 8715.3, NASA General Safety Program Requirements
- f. [DPD-1000.1-001](#), Governance and Strategic Management Handbook
- g. [DPD-8700.1-001](#), Organizational & Individual Safety Responsibilities

P.4 Applicable Documents

- a. NPR 8715.3, NASA Procedural-NASA General Safety Program Requirements
- b. NPR 7150.2, NASA Software Engineering Requirements
- c. NPR 8621.1, NASA Procedural Requirements for Mishap and Close Call Reporting, Investigating, and Recordkeeping
- d. [DPD-1000.1-001](#), Governance and Strategic Management Handbook
- e. [DPD-2800.2-001](#), Managing Information Technology (IT)
- f. [DPR-7123.1-001](#), Systems Engineering Requirements Document
- g. [DPR-7123.2-001](#), Waivers and Deviations to Technical Requirements and Standards

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

- h. NASA-STD-8719.13, Software Safety Standard
- i. NASA-STD-8739.8, Software Assurance Standard
- j. NASA-HDBK-4008, Programmable Logic Devices (PLD) Handbook
- k. NASA-HDBK-8739.23, NASA Complex Electronics Handbook For Assurance Professionals
- l. [DCP-S-002](#), Hazard Management Procedure
- m. [DCP-S-007](#), Software Assurance
- n. [DCP-X-009](#), Airworthiness and Flight Safety Review Process
- o. Requirements and Standards ISO 24765:2010 system and software engineering - vocabulary
- p. RTCA DO-178, Software Considerations in Airborne Systems and Equipment Certification

P.5 Measurement/Verification

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

P.6 Cancellation

DPR-7150.2-001, Baseline-1, Software Engineering Requirements, dated June 3, 2010

David McBride, Center Director

Date

CHAPTER 1: INTRODUCTION

1.1 Sources of Requirements

1.1.1 This document seeks to provide a unified set of process requirements for software development/management activities at the Center. It includes the tailored software engineering requirements specified in NPR 7150.2, and the software safety requirements specified in NASA-STD-8719.13. Finally, it includes requirements derived from RTCA DO-178, Software Considerations in Airborne Systems and Equipment Certification, and Center unique to fill in those areas where Center processes need to provide more stringent requirements to support airworthiness.

1.2 Document Scope

1.2.1 The requirements of this document cover software created or acquired by or for the Center, including COTS, GOTS, MOTS, open source, reuse, legacy, and heritage software

1.2.2 The requirements found in this document apply to specific systems and subsystems as determined through the use of the software classifications described in Section 3, in conjunction with the Requirements Mapping Matrix found in Appendix E.

1.2.3 The requirements found in this document and the classification system used to levy these process requirements represent the Centers tailored approach to interpreting the NASA requirements specified in NPR 7150.2 and NASA-STD-8719.13.

1.2.4 The requirements found in this document shall be applied to software development, maintenance, operations, retirement, management, acquisition, and assurance activities started after its effective date of issuance. Contracts that involve software development will include references to this DPR.

1.2.5 This document provides procedural requirements to the responsible project managers and contracting officers for NASA contracts. It is made applicable to contractors through contract clauses, specifications, or statements of work in conformance with the NASA Federal Acquisition Regulation (FAR) Supplement.

1.2.6 The requirements found in this document do not supersede more stringent requirements imposed by individual NASA organizations and other Federal Government agencies.

1.2.7 Any material not identified by a "shall" in this document is informative in nature (e.g., notes, introductory text, etc.).

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

1.3 Description of Software

1.3.1 For the purposes of this document, software is defined in NPR 7150.2, NASA Software Engineering Requirements, Section A.30, as “Computer programs, procedures, scripts, rules, and associated documentation and data pertaining to the development and operation of a computer system.” Software includes programs and data. This also includes COTS, GOTS, MOTS, reused software, auto generated code, embedded software, firmware, and open source software components.

1.3.2 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. Embedded software: Software that is part of a larger system and performs some of the requirements of that system.
- d. 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.
- e. Firmware: Combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device.
- f. Previously developed software: Software that has been produced prior to or independent of the project’s software development plan including software that is obtained or purchased from outside sources.
- g. Reusable software product: 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.
- h. Software tool: A computer program used in the development, testing, analysis, or maintenance of a program or its documentation.
- i. Support software: Software that aids in the development or maintenance of other software.
- j. System software: Software designed to facilitate the operation and maintenance of a computer system and associated programs. Reference: System and Software Engineering Vocabulary (ISO 24765).

Before use, check the Master List to verify that this is the current version.

This document may be distributed outside of the Center.

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

1.3.4 NASA has developed handbooks to address assurance of complex electronics. These include NASA-HDBK-8739.23, NASA Complex Electronics Handbook For Assurance Professionals, as well as NASA-HDBK-4008, Programmable Logic Devices (PLD) Handbook. The Center is in the process of assessing the need for stand-alone processes for the development/integration of these components.

CHAPTER 2: RESPONSIBILITIES

1. The Designated Governance Framework for the Center is defined in DPD-1000.0-001. Chapter 6 of NPR 7150.2 provides the tailoring, engineering technical authority, and compliance requirements. The requirements mapping matrix, found in Appendix D identifies the technical authority (TA) for each of the NPR 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 Safety and Mission Assurance office. This office ensures that the requirements found in NPR 8715.3 are being met by the Center. The Center has defined these roles and responsibilities in [DPD-8700.1-001](#), Organizational & Individual Safety Responsibilities. This DPR uses the information found in DPD-8700.1-001 to allocate responsibility for each requirement defined to the appropriate organizational level. This allocation of responsibility starts with the Center Director and is delegated down to the following organizations:

- a. Director of Mission Support,
- b. Director for Research Engineering
- c. Director for Mission Information and Test Systems,
- d. Director for Flight Operations
- e. Director, Safety and Mission Assurance,
- f. Acquisition Management Officer.

2. The general delegation strategy is listed below. The specific mapping of requirements to Center TA delegates is provided in Appendix D.

2.1 Armstrong Center Director

2.1.1 The Center Director is the Designated Governing Authority for Center level requirements dealing with applicability and scope, best practices, expertise of TA, organizational capability, tailoring of requirements, and training. The Center Director is also the Designated Governing Authority for requirements covering legal compliance. See Appendix D for the specific list.

2.2 Director for Mission Support

2.2.1 The Director for Mission Support is the Designated Governing Authority for the requirements in this document that deal with the development, purchase, usage, and/or maintenance of software within the Facilities & Asset Management, Chief Financial Officer, and Acquisition Management areas. Specific categories include:

- a. Compliance
- b. Project formulation

- c. Software life cycle
- d. Software plans
- e. Software requirements
- f. Software design
- g. Peer reviews/inspections
- h. Software implementation
- i. Software testing
- j. Software verification and validation
- k. Software configuration
- l. Software measurement
- m. Software operations, maintenance, and retirement

See Appendix D for the specific lists.

2.3 Director for Research and Engineering

2.3.1 The Director for Research and Engineering is the Designated Governing Authority for the requirements in this document that deal with the development, purchase, usage, and/or maintenance of software within the various Research and Engineering branches. Specific categories include:

- a. Compliance
- b. Project formulation
- c. Software life cycle
- d. Software plans
- e. Software requirements
- f. Software design
- g. Peer reviews/inspections
- h. Software implementation
- i. Software testing
- j. Software verification and validation
- k. Software configuration
- l. Software measurement
- m. Software operations, maintenance, and retirement

2.3.2 The Director for Research and Engineering is also the Designated Governing Authority for the center-wide software training requirements listed in this document.

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

See Appendix D for the specific lists.

2.4. Director for Mission Information and Test Systems

2.4.1 The Director for Mission Systems is the Designated Governing Authority for the requirements in this document that deal with the development, purchase, usage, and/or maintenance of software within the Chief Information Officer (CIO) and Mission Information and Test Systems areas. Specific categories include:

- a. Compliance
- b. Project formulation
- c. Software life cycle
- d. Software plans
- e. Software requirements
- f. Software design
- g. Peer reviews/inspections
- h. Software implementation
- i. Software testing
- j. Software verification and validation
- k. Software configuration
- l. Software measurement
- m. Software operations, maintenance, and retirement

See Appendix D for the specific lists.

2.5 Director for Flight Operations

2.5.1 The Director for Flight Operations is the Designated Governing Authority for the requirements in this document that deal with the development, purchase, usage, and/or maintenance of software within the Flight Operations Directorate. Specific categories include:

- a. Compliance
- b. Project formulation
- c. Software life cycle
- d. Software plans
- e. Software requirements
- f. Software design
- g. Peer reviews/inspections

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

- h. Software implementation
- i. Software testing
- j. Software verification and validation
- k. Software configuration
- l. Software measurement
- m. Software operations, maintenance, and retirement

2.5.2 The Director for Flight Operations is also the Designated Governing Authority for requirements covering the control of software loaded on aircraft.

See Appendix D for the specific lists.

2.6 Director for Safety and Mission Assurance

2.6.1 The Director for Safety and Mission Assurance is the Designated Governing Authority for the requirements in this document that deal with the development, purchase, usage, and/or maintenance of software within the Safety and Mission Assurance Organization. Specific categories include:

- a. Compliance
- b. Project formulation
- c. Software life cycle
- d. Software plans
- e. Software requirements
- f. Software design
- g. Peer reviews/inspections
- h. Software implementation
- i. Software testing
- j. Software verification and validation
- k. Software configuration
- l. Software measurement
- m. Software operations, maintenance, and retirement

2.6.2 The Director for Safety and Mission Assurance is also the Designated Governing Authority for all software safety requirements derived from NASA-STD-8719.13.

See Appendix D for the specific lists.

2.7 Acquisition Management Officer

2.7.1 The Acquisition Management Officer is the Designated Governing Authority for software contract requirements listed in this document.

See Appendix D for the specific list.

CHAPTER 3. SOFTWARE CLASSIFICATION

1. Appendix E of NPR 7150.2 defines eight classifications for NASA software. These classifications are based on:

- a. Usage within a NASA system,
- b. Criticality of the system to NASA's programs and projects,
- c. Extent to which humans depend on the system,
- d. Developmental and operational complexity, and
- e. The extent of the Agency's investment.

2. These definitions are assigned an alphabetic classification identification from A-H. Class A-E covers engineering software while class F-H cover business and IT software. In addition, the NPR identifies an additional test applied to software defined as class A-E. This is the software safety litmus test (Reference NASA-STD-8719.13, Appendix A). The results of this litmus 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-8719.13. 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 "Software Considerations in Airborne Systems and Equipment Certification" (RTCA DO-178). The historical software classification system used at the Center was defined in [DCP-S-007](#), Software Assurance. The classification method used in this DPR 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.

Note: While this DPR provides the necessary hooks to allow for the inclusion of business and IT infrastructure software classifications defined in NPR 7150.2, acquirers/developers of these classes of software may find that the classification methodology defined in the NPR to be a better match for their particular needs. If this is the case, the NPR classification and requirements may be substituted for those found in this DPR without the need for generating a waiver. This is not the case for flight and ground software used to support that falls outside of the definitions associated with business and IT infrastructure software. The reason for this is that this DPR levies additional requirements that, in some cases, exceed the requirements levied in NPR 7150.2. While projects may choose to work to the software classifications/requirements defined in NPR 7150.2, they will need to generate and submit waivers documenting these deviations. Waivers/deviations need to be submitted in accordance with [DPR-7123.2-001](#), Waivers and Deviations to Technical Requirements and Standards.

Before use, check the Master List to verify that this is the current version.

This document may be distributed outside of the Center.

3.1 General

3.1.1 Requirements in this document are assigned to software items according to the criticality of that software. Specifically, software is grouped into 1 of 4 different classifications based on the most severe consequence of a software-controlled event. These classifications are closely coupled to the hazard categories described in [DCP-S-002](#). Specifically, these categories are as follows.

- a. Class I: Catastrophic
- b. Class II: Critical
- c. Class III: Minor
- d. Class IV: Negligible

3.1.1.1 The use of Roman numerals is meant to reduce confusion with other software standards such as NPR 7150.2 and RTCA DO-178 that use alphabetic classifications.

3.1.2 Since this document 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.

3.1.3 Identification/incorporation the safety critical software increases the number of requirements levied by NPR 7150.2. It also levies additional requirements called out in NASA-STD-8719.13. 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. See Sections 3.4 and 3.5 for more specific definitions.

3.2 The Classification Process

3.2.1 The criticality of a software item should be determined using a Preliminary Hazard Analysis (PHA) 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 configuration management 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 must be developed to the more stringent standards.

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

3.2.2 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.

3.2.3 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 need to consult with Center management to discuss the software engineering 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

3.3.1 Safety critical is defined in NPR 8715.3, as “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.”

3.3.2 Software is considered safety-critical if it resides on a safety critical system and meets the criteria defined in Appendix A of NASA-STD-8719.13.

3.3.3 In accordance with [DPD-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. The Center or its contractor personnel and its equipment are at risk,
- b. The Center has an assigned safety responsibility (i.e., flight, ground, range, environmental, etc.), or
- c. The Center owns the asset and are not otherwise excluded by agreement or contract.

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

3.4 Software Classifications – Safety Critical

3.4.1 Software considered safety critical using the definition in Section 3.3 is further classified based on the most severe consequence of a software-controlled event. The classification criteria are as follows:

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

a. Class I-S: Catastrophic

- (1) Death or permanently disabling/life-threatening injury, or loss of crew
- (2) Destruction of facility on the ground, major system, vehicle, termination of project.

b. Class II-S: Critical

- (1) Severe/lost time injury or occupational illness
- (2) Major loss/damage to facility, system, equipment, flight hardware, vehicle

c. Class III-S: Moderate Not Applicable

Note: The previous version of DPR 7150.2 defined Class III-S as “Medical treatment for a minor injury or occupational illness (no lost time.)” SWE-133 states that software safety criticality shall be performed in accordance with NASA-STD-8739.3. The first requirement is that software must reside in a safety critical system. The definition safety critical states: “Any condition, event, operation, process, equipment, or system that could cause or lead to severe injury...” Given these definitions a software classification of III-S cannot exist.

d. Class IV-S: Minimal Not Applicable**3.5 Software Classifications – Non-Safety Critical**

3.5.1 Software not considered safety critical using the definition in Section 2.2 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
- (2) Recovery/replacement cost equal to or greater than \$2M

b. Class II: Critical

- (1) Long term project delay
- (2) Loss of some project critical data
- (3) Recovery/replacement cost equal to or greater than \$500K, but less than \$2M
- (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 noncritical project data
- (3) Minor loss/damage to facility, system, equipment, or flight hardware
- (4) Recovery/replacement cost equal to or greater than \$50K, but less than \$500K.
- (5) Interruptions in the availability of critical data
- (6) Center-wide productivity impact

d. Class IV: Minimal

- (1) Productivity impact to small number of users

Note: The monetary values for recovery/replacement costs are found in DCP-S-002. The costs found in this DPR should be used only as a reference. If a discrepancy exists between the specified recovery/replacement costs those found in this document, DCP-S-002 take precedence.

3.6 Classification Guidelines

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

a. Destruction of facility, major system, or vehicle

(1) The intent of this statement is to capture consequences that would likely lead to a NASA Class A Mishap per NPR 8621.1, Figure 1: hull loss of a crewed aircraft or greater than \$2M in property damage to a facility or system. In some cases, however, loss of a test article is either planned or anticipated and thus may not drive software criticality to the highest level. Examples include:

- (a) Intentional destruction of a vehicle
- (b) Vehicles or systems not intended to be recovered once the test is complete.

b. Recovery/replacement costs

(1) [DCP-S-002](#) and NPR 8621.1 both provide criteria for recovery/replacement costs (for instance, \$2M is the threshold for a Category I Hazard in DCP-S-002, and a Type A Mishap in NPR 8621.1). NPR 8621.1, Section 1.3.3, provides guidance as to how to make that assessment.

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

c. Major Damage vs. Destruction

(1) 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

(1) When software falls into the category of business and IT infrastructure (as defined in NPR 7150.2, Appendix E) then it should be classified in accordance with the guidance provided by the Center CIO. The CIO may decide to apply the business and IT infrastructure software classifications (F-H) found in NPR 7150.2 in lieu of the classifications defined in this chapter of the DPR.

e. Long term delay

(1) The definition of long term delay must also be 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 mission

(1) 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 by definition 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 document, loss of mission implies that there will be other opportunities to collect the data.

g. Interruptions in Availability

(1) 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 Architectural Considerations

3.7.1 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).

Before use, check the Master List to verify that this is the current version.

This document may be distributed outside of the Center.

b. These mitigations meet the requirements levied in NPR 8715.3, NASA General Safety Program Requirements, Section 1.8.

c. These mitigations are verifiable and verified.

3.7.2 Note that 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.	Can be considered when classifying the criticality of the software.
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. SOFTWARE ENGINEERING REQUIREMENTS

1. This section includes the specific Center software engineering requirements as levied by this standard. Requirements that flow down from other NASA documents (NPR 7150.2 and NASA-STD-8719.13, specifically) are not reprinted here, but are instead referenced by number. Please refer to the parent document for the actual text.

2. The applicability of requirements in this chapter is a function of the classification of the software as determined in Chapter 3. Appendix D contains the cross-reference matrix showing which requirements apply to each software class.

4.1 Center Level Software Engineering Requirements

4.1.1 Applicability and Scope

a. (R.0060) Effective date – SWE-001

4.1.2 Organizational Capability and Improvement

a. (R.0070) Center plan – SWE-003

b. (R.0080) SW processes – SWE-005

c. (R.0590) Intent of Capability Maturity Model Integration (CMMI) – SWE-032

Note: See Appendix F for the Center approach for meeting the intent of CMMI.

4.1.3 Best Practices

a. (R.0100) Identify applicable practices – SWE-099

4.1.4 Training

a. (R.0110) Software engineering training – SWE-100

b. (R.0120) Software training plan – SWE-101

4.1.5 Software Plans

a. (R.0130) Center software training plan – SWE-107

b. (R.0140) Center software engineering improvement Plan – SWE-108

4.1.6 Tailoring of Requirements

a. (R.0150) Alternate requirement request – SWE-120

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

- b. (R.0160) Document approved alternate requirements – *SWE-121*

4.1.7 Expertise of TA(s)

- a. (R.0170) Non-IT and non-business – *SWE-122*
- b. (R.0181) Elevate disagreements with software safety criticality assessment (SSCA) – *SSS-007*

4.1.8 Compliance

- a. (R.0190) Direction for TA – *SWE-124*
- b. (R.0200) Considerations for waivers – *SWE-126*
- c. (R.0210) Review of "P(Center)" – *SWE-127*
- d. (R.0220) Compliance records – *SWE-128*
- e. (R.0221) Compliance with NPR requirements – *SWE-139*
- f. (R.0222) Documenting Center tailoring of NPR requirements - *SWE-140*

4.1.9 Software Safety Requirements

4.1.9.1 Determination of Safety-Critical Software

- a. (R.0231) Acquirer complies with software safety standard – *SSS-002*
- b. (R.0233) Acquirer SMA approves provider's SSCA – *SSS-006*

4.1.9.2 Certification Process

- a. (R.0232) Acquirer imposes software safety standard on provider - *SSS-003*
- b. (R.0241) Provider SMA approves safety critical docs - *SSS-009*
- c. (R.0242) Acquirer SMA approves docs with safety requirements - *SSS-010*
- d. (R.0261) Provider collects safety objective evidence for acquirer acceptance - *SSS-061*
- e. (R.0341) Provider adheres to software safety standard - *SSS-004*

f. (R.0342) Provider obtains acquirer and SMA approval for safety evidence of acceptance - SSS-062

4.1.9.3 Operational Use of Software

a. (R.0361) Provider addresses safe decommissioning of system in retirement plan - SSS-065

b. (R.0371) Acquirer and provider SMA concurs on safe decommissioning of system in retirement plan -SSS-066

4.1.9.4 Waivers/Deviations

a. (R.0381) Acquirer SMA maintains copy of waivers/deviations - SSS-017

4.1.10 Center Support of Headquarters

a. (R.0010) Headquarters funds software engineering - SWE-002

b. (R.0020) Headquarters benchmarks - SWE-004

c. (R.0030) Headquarters maintains list of projects containing software - SWE-006

d. (R.0040) Headquarters maintains process asset library - SWE-098

e. (R.0050) Headquarters authorizes appraisals - SWE-129

4.2 Project Level Software Engineering Requirements

4.2.1 Compliance with Laws, Policies, and Requirements

a. (R.0090) Software safety - SWE-023

4.2.2 Software Life Cycle Planning

a. (R.0450) Software plan - SWE-013

b. (R.0460) Execute plan - SWE-014

c. (R.0470A) Cost estimation - SWE-015

d. (R.0480A) Schedule - SWE-016

e. (R.0490) Training - SWE-017

f. (R.0500) Reviews - SWE-018

- g. (R.0510) Life Cycle - SWE-019
- h. (R.0520) Software classification - SWE-020
- i. (R.0530) Software classification changes - SWE-021
- j. (R.0540A) Software assurance - SWE-022
- k. (R.0550) Plan tracking - SWE-024
- l. (R.0560) Corrective action - SWE-025
- m. (R.0570) Changes - SWE-026

4.2.3 Commercial, Government, and Modified Off-The-Shelf Software

- a. (R.0580) COTS, GOTS, MOTS – SWE-027

4.2.4 Project Formulation Requirements

- b. (R.0600) Supplier selection – SWE-035
- c. (R.0610) Acquisition planning - SWE-038

4.2.5 Software Safety Requirements

- a. (R.0531) Independent classification assessment - SWE-132
- b. (R.0532) Develop software safety criticality assessment - SWE-133
- c. (R.0681) Provider, with SMA, includes software in system safety analysis - SSS-020
- d. (R.0691) Acquirer, with SMA, develops and maintains software safety plan – SSS-031
- e. (R.0692) Provider includes list of software safety plan contents - SSS-037

4.3 System Level Software Engineering Requirements

4.3.1 Software Verification and Validation

- a. (R.0740) Verification planning - SWE-028
- b. (R.0750) Validation planning - SWE-029

c. (R.0760) Verification results - SWE-030

d. (R.0770) Validation results - SWE-031

4.3.2 Project Formulation Requirements

a. (R.0780) Options for acquisition - SWE-033

b. (R.0790) Acceptance criteria - SWE-034

c. (R.0800) Software processes and tasks - SWE-036

d. (R.0810) Milestones - SWE-037

4.3.3 Software Requirements

a. (R.0820) Documented requirements - SWE-049

b. (R.0830) Software requirements - SWE-050

c. (R.0840A) Flow-down and derived requirements - SWE-051

d. (R.0850) Bidirectional trace - SWE-052

e. (R.0860) Manage requirements change - SWE-053

f. (R.0870) Corrective action - SWE-054

g. (R.0880) Requirements validation - SWE-055

h. (R.0882) High Level Algorithm Review: The project shall review the proposed high level algorithm(s) to ensure their accuracy and behavior, especially in the area of discontinuities. Ref DO-178, 6.3.1g

i. (R.0924) Architecture review vs. high level requirements: The project shall review the proposed software architecture to ensure it does not conflict with the high-level requirements, especially functions that ensure system integrity, for example, partitioning schemes. Ref DO-178, 6.3.3a

j. (R.0926A) Consistency: The project shall ensure that a correct relationship exists between the components of the software architecture including data flow, control flow and interfaces. Ref DO-178, 6.3.3.3b

k. (R.0928) Review of software architecture partitioning: The project shall review the proposed software architecture to ensure that partitioning breaches are prevented or isolated. Ref DO-178, 6.3.3c

4.3.4 Software Design

- a. Bidirectional trace (R.0890) – See NPR 7150.2, SWE-059
- b. Document design (R.0900) – See NPR 7150.2, SWE-056
- c. Architecture (**R.0910**) – See NPR 7150.2, SWE-057
- d. Detailed design (**R.0920**) – See NPR 7150.2, SWE-058
- e. Low Level Algorithm Review: The project shall review the proposed low level algorithm(s) to ensure their accuracy and behavior, especially in the area of discontinuities. (**R.0922**) – (DO-178, 6.3.2g)

4.3.5 Software Implementation

- a. (R.0930A) Maintain traceability - SWE-064
- b. (R.0940A) Coding standards - SWE-061
- c. (R.0950) Unit test - SWE-062
- d. (R.0960) Version description - SWE-063

4.3.6 Software Testing

- a. (R.0970A) Models, simulations, tools - SWE-070
- b. (R.0980) Plan, procedures, reports - SWE-065
- c. (R.0990) Perform testing - SWE-066
- d. (R.1000) Test for compliance - SWE-067
- e. (R.1001) Static analysis tools - SWE-135
- f. (R.1002A) The project shall perform structural coverage analysis after requirement verification is performed to identify code, including interfaces that were not exercised during that testing. Ref DO-178, 6.4.4.2
- g. (R.1004A) The project shall use the results of the structural coverage analysis to identify and perform additional software/interface

(1) every decision in the program has taken all possible outcomes at least once, testing to ensure that

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

(2) every condition in a decision in the program has taken all possible outcomes at least once, and

(3) every condition in a decision has been shown to independently affect that decisions outcome. Ref DO-178, 6.4.4.3

h. (R.1006A) The project shall identify and remove any extraneous/dead code identified during the structural coverage analysis and assess the effect and the need for reverification based on that change.

Note: If extraneous code is found at the source code or object code level, it may remain if analysis shows it does not exist in the executable object code (for example, due to smart compiling, linking, or some other mechanism). Ref DO-178, 6.4.4.3c

i. (R.1008) The project shall identify any deactivated code (code that is not intended to be executed in any expected operational configuration) and perform analysis and/or testing to show that the means by which any such code could be inadvertently executed are prevented, isolated, or eliminated. Ref DO-178, 6.4.4.3d

j. (R.1010) Evaluate test results - SWE-068

k. (R.1020) Document defect and track - SWE-069

l. (R.1030) Update plans and procedures - SWE-071

m. (R.1040) Maintain Traceability - SWE-072

n. (R.1050) Platform or High-Fidelity simulation. - SWE-073

4.3.7 Software Operations, Maintenance, and Retirement

a. (R.1060) Document maintenance plans - SWE-074

b. (R.1070) Plan operations, maintenance, and retirement - SWE-075

c. (R.1080) Implement plans - SWE-076

d. (R.1090) Deliver software product - SWE-077

e. (R.1100) As-built documentation - SWE-078

4.3.8 Peer Reviews/Inspections

a. (R.1110) Requirements and test plans - SWE-087

Before use, check the Master List to verify that this is the current version.

This document may be distributed outside of the Center.

- b. (R.1120) Checklist, criteria, and tracking - SWE-088
- c. (R.1121) Reporting results - SWE-137
- d. (R.2740A) Software inspection/peer review - SWE-119
- e. (R.3140A) Basic measurements - SWE-089

4.3.9 Software Safety Requirements

4.3.9.1 Determination of Safety-Critical Software

- a. (R.1131) Acquirer SMA performs SSCA Part 1 - SSS-001
- b. (R.1132) Provider SMA performs SSCA - SSS-005
- c. (R.1171) Provider maintains SSCA - SSS-008

4.3.9.2 Program, Project, Facility Management

- a. (R.1191) Provider, with SMA, develops, baselines, and configuration manages software safety plan per contract/MOU/MOA - SSS-035

4.3.9.3 Off-the-shelf Software (OTS)

- a. (R.1211) Provider, with SMA, performs safety analysis on OTS and reused software SSS-019

4.3.9.4 Waivers/Deviations

- a. (R.1251) Prepare waiver/deviation package if any requirements cannot be met SSS-015
- b. (R.1261) SMA evaluates and submits waiver/deviation package - SSS-016

4.3.9.5 Software Safety Requirements and Analysis

- a. (R.1271) Software safety modes and states - SWE-134
- b. (R.1291) Provider identifies/tags for software safety requirements - SSS-038
- c. (R.1292) Provider SMA concurs on software safety requirements identification/tagging - SSS-041

d. (R.1321) Provider includes hardware/software/operator safety constraints in requirements document - SSS-039

4.3.9.6 Software Design and Safety Analysis

a. (R.1341) Provider unique tags software design implementing safety features/methods - SSS-043

4.3.9.7 Software Test and Safety Analysis

a. (R.1401) Provider verifies software safety requirements by test - SSS-054

b. (R.1411) Provider performs testing verifying hazards are eliminated/controlled SSS-055

c. (R.1421) Provider includes unit/component testing for safety aspects of software SSS-056

d. (R.1451) Provider SMA concurs software safety verification results - SSS-053

e. (R.1461) Provider SMA concurs on test plans/procedures - SSS-051

f. (R.1462) Provider includes software conditions impacting performance in system testing - SSS-052

g. (R.1501) Provider verifies/validates by tests including loads, stress, and off-nominal conditions - SSS-057

h. (R.1511) Provider verifies by tests including correct and safe operations, transitioning to safe state - SSS-058

i. (R.1531) Provider, with SMA, verifies by analysis, inspection, or demonstration if cannot by test - SSS-059

4.4 Developer Level Software Engineering Requirements

4.4.1 Software Implementation

a. (R.1580) Design→code - SWE-060

4.5 System Safety Software Engineering Requirements

4.5.1 Risk Management

a. (R.1620) Continuous risk management – SWE-086

4.5.2 Software Safety Requirements

4.5.2.1 Software and System Safety

- a. (R.1661) Provider identifies new or updated software safety requirements - SSS-023

4.5.2.2 Software Safety Personnel

- a. (R.1731) Acquirer SMA verifies safety contents in contract/MOU/MOA - SSS-033
- b. (R.1781) Acquirer assigns SMA as approving member - SSS-021

4.5.2.3 Software Safety Planning

- a. (R.1811) Acquirer SMA evaluates and approves provider software safety plan SSS-034
- b. (R.1812) Develop software safety plan - SWE-130
- c. (R.1813) Generate plan - SWE-138

4.5.2.4 Traceability

- a. (R.1851) Provider traces software safety requirements to system hazards - SSS-011
- b. (R.1871) Provider SMA verifies configuration management implementation - SSS-013
- c. (R.1881) Provider SMA concurs with trace results - SSS-012

4.5.2.5 Software Safety Requirements and Analysis

- a. (R.1831) Provider SMA reevaluates SSCA, obtains acquirer SMA approval for changes - SSS-024
- b. (R.1941) Provider develops and maintains evidence of software safety requirements compliance - SSS-022
- c. (R.1942) Provider SMA analyzes the software safety requirements per criteria SSS-040
- d. (R.2011) Provider updates software requirements analysis results in system safety data package. -SSS-042
- e. (R.2041) Provider SMA performs software design analysis per criteria -SSS-044

f. (R.2131) Provider updates system safety data package based on software design analysis results -SSS-045

g. (R.2222) Provider SMA concurs on code hazards/controls and updated data products - SSS-050

4.5.2.6 Software Design and Safety Analysis

a. (R.2121) Provider SMA concurs on design hazards/controls and updated data products - SSS-046

4.5.2.8 Software Test and Safety Analysis

a. (R.2331) Provider SMA verifies any new hazards updated in system safety data package - SSS-060

4.5.2.9 Operational Use of Software

a. (R.2371) Provider SMA verifies regression testing required for implementing new requirements/changes -SSS-063

b. (R.2391) Provider SMA evaluates user manuals and procedures for safe operation/impacts -SSS-064

4.6 Configuration Management Requirements

4.6.1 Software Configuration Management (CM)

a. (R.2400) Develop CM plan - SWE-079

b. (R.2410) Track and evaluate changes - SWE-080

c. (R.2420A) Identify software configuration items - SWE-081

d. (R.2430) Authorize changes - SWE-082

e. (R.2440) Maintain records - SWE-083

f. (R.2450A) Perform configuration audits - SWE-084

g. (R.2460) Implement procedures - SWE-085

4.6.1.1 Loading Flight Software

- a. (R.2461) Labeling of software media: All software media (tape, disk, or chip) shall be identified and physically labeled at the time of production. AFRC unique (flight software only)
- b. (R.2462) Version description document (VDD): Prior to installation on the aircraft, a VDD shall be produced. AFRC unique (flight software only)
- c. (R.2463) Flight media release (FMR) form included in version description document: The VDD shall contain a FMR Form 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. AFRC unique (flight software only)
- d. (R.2464) Flight software installation procedure: A procedure shall be written for flight software installation into the aircraft computer and for verification of correct loading. AFRC unique (flight software only)
- e. (R.2465) Specification of flight(s) on flight media release form: Flight software for a specific flight or block of flights shall be designated by the software manager on a FMR form. AFRC unique (flight software only)
- f. (R.2466) Confirmation of correct software version before flight: Quality inspection shall verify the correct flight software is loaded for the specified flight according to approved procedures. AFRC unique (flight software only)

4.6.2 Software Safety Requirements

- a. (R.2491) Provider evaluate discrepancy reports for safety impacts - SSS-029
- b. (R.2501) Provider SMA concurs on all software changes - SSS-028
- c. (R.2511) Provider traces discrepancies/problems/failures to system hazards - SSS-030
- d. (R.2521) Provider SMA analyzes software changes for safety impacts - SSS-027

4.7 Documentation Requirements

4.7.1 Software Plans

4.7.1.1 Configuration Management Plan

- a. (R.2610) Software CM plan - SWE-103

4.7.1.2 Software Assurance Plan

- a. (R.2640A) Software assurance plan - SWE-106

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

4.7.1.3 Software Development Plan

- a. (R.2600) Software development/management Plan - SWE-102

4.7.1.4 Software Maintenance Plan

- a. (R.2630A) Software maintenance plan - SWE-105

4.7.1.5 Test Plans

- a. (R.2620) Software test plan - SWE-104

4.7.2 Software Design Documents

- a. (R.2650) Software requirements specification - SWE-109
- b. (R.2660A) Software data dictionary - SWE-110
- c. (R.2670) Software design description - SWE-111
- d. (R.2680A) Interface design description -SWE-112
- e. (R.2690A) Software change request/problem report - SWE-113
- f. (R.2700A) Software test procedures - SWE-114
- g. (R.2710A) Software user manual - SWE-115
- h. (R.2720) Software version description - SWE-116

4.7.3 Software Reports

- a. (R.2730A) Software test report - SWE-118

4.7.4 Compliance

- a. (R.2750) Compliance matrix - SWE-125

4.7.5 Software Safety Requirements

4.7.5.1 Software Development Plan

- a. (R.2771) Tool accreditation - SWE-136
- b. (R.2781) Provider determines if tools impact safety - SSS-018

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

4.7.5.2 Other Documentation Requirements

- a. (R.2861) Develop IV&V execution plan as needed -SWE-131

4.8 Statement of Work Content Requirements

4.8.1 Software Contract Requirements

- a. (R.2990A) Source code access -SWE-042
- b. (R.3000) Software measurement data - SWE-044
- c. (R.3010) Insight into test - SWE-039
- d. (R.3020A) Electronic access - SWE-040
- e. (R.3030A) Open source - SWE-041
- f. (R.3040) Track change request - SWE-043
- g. (R.3050) Joint audits - SWE-045
- h. (R.3060) Software schedule - SWE-046
- i. (R.3070A) Traceability data - SWE-047
- j. (R.3080) Solicitation - SWE-048

4.8.2 Software Safety Requirements

4.8.2.1 Contract Management

- a. (R.3101) Acquirer identifies additional software safety requirements, includes them in contract/MOU/MOA - SSS-032
- b. (R.3102) Acquirer SMA evaluates and approves provider software safety plan - SSS-036
- c. (R.3131) Provider assigns SMA as voting member of provider software change control process - SSS-025
- d. (R.3132) Acquirer assigns SMA as voting member of acquirer software change control process - SSS-026

4.9 Metrics Requirements

4.9.1 Software Measurement

- a. (R.3150) Software measurement areas - SWE-091
- b. (R.3160) Collection and storage - SWE-092
- c. (R.3170) Analyze data - SWE-093
- d. (R.3180) Report analysis - SWE-094
- e. (R.3220) Objectives - SWE-090

4.9.2 Software Report Requirements

- a. (R.3230A) Software metrics report - SWE-117

4.9.3 Software Measurement Requirements

- a. (R.3190) Mission directorates define software measurement system - SWE-095
- b. (R.3200) Mission directorates define software measurement objectives - SWE-096

CHAPTER 5. COMPARISON TO NPR 7150.2

5.1 Overview

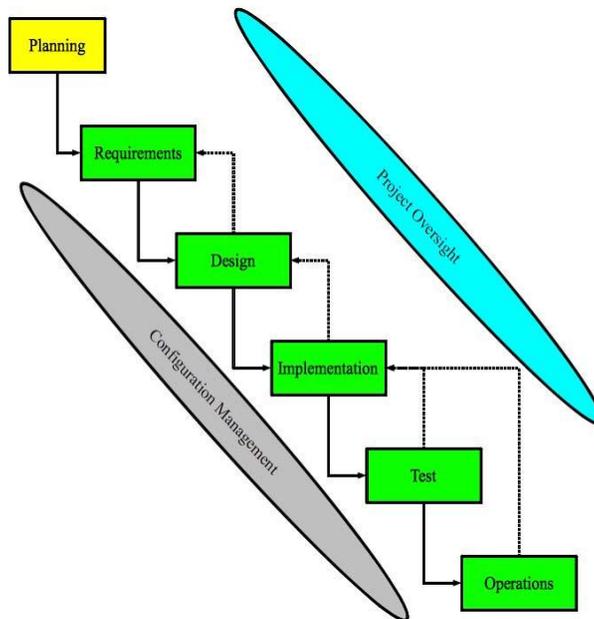
5.1.1 The following section describes those areas where the Center requirement exceeds the requirements in NPR 7150.2. Note that requirements derived from other sources (NASA-STD-8739.13, [DCP-S-007](#), or RTCA DO-178) are not listed.

5.1.2 Software components are part of larger systems/subsystems that contain them. As a result, the software engineering lifecycle needs to support the larger systems engineering lifecycle that contains it. The life cycle models do not need to match each other but they do need to be defined and work in concert in order to achieve the desired results. This DPR does not attempt to mandate a software engineering lifecycle model. Projects are encouraged to reference [DPR-7123.1-001](#), Procedural Requirements for Systems Engineering, then tailor their systems/software engineering life cycle model that suits their needs, while maintaining compliance with DPR-7123.1-001. This chapter does identify software engineering artifacts defined in NPR 7150.2, NASA-STD-8719.13, as well as those internally defined within the DPR. These artifacts have been placed in the following categories:

- a. Planning
- b. Requirements
- c. Design
- d. Implementation
- e. Test
- f. Operations
- g. Configuration management
- h. Life cycle independent

This life cycle flow can be found in figure 5.1

Figure 5.1: Life Cycle Model



5.2 Planning

5.2.1 Planning activities need to be performed prior to the start of the software development lifecycle. The objective of the planning effort is to define the approach to be taken to produce/maintain/operate the software being developed to support the product. The artifacts/products shown in table 5.1 should be produced as part of this process.

Table 5.1: Documentation Requirements for Planning Phase

Req	Document	I	II	III	IV	S	Remarks	Content Definition
R.0460	Software Development Plans	X	X	X	X	X	Content defined for all but Class IV	R.2600
R.0460	Software Management Plans						May be combined with software development plan	R.2600
R.0460	Software Configuration Management Plans	X	X	X	X	X	Content defined for all but Class IV. May be combined with project configuration management plan	R.2610
R.2640A	Software Assurance Plans	X	X	*	*	X	Content defined for all but Class III and IV. May be performed at Class III, IV at request of SMA.	R.2640A
R.1171	Software Safety Critical Assessment	X	X	*	*	X	Independent classification assessment needed by SMA	
R.0692	Software Safety Plans	X	X	*	*	X	Only required if safety critical CSCIs are identified	R.0691
R.2861	IV&V Project Execution Plan	X	X	*	*	X	Only required if selected for IV&V	

5.3 Requirements

5.3.1 Requirements activities are normally performed at the start of, or early in, the software engineering lifecycle. The objective of the requirements effort is to define/decompose a meaningful set of requirements that can be used to design/develop/implement the software product. Since software is a component of a higher level system, there will be a flow down of information/requirements from the higher level system. The information flowed down may include, but are not limited to, needs, goals, objectives, operational modes/states, interfaces, and requirements. This information is further defined/decomposed to define the objectives, requirements, and high level software architecture. This information is then documented and reviewed prior to starting formal software design/development. The artifacts/products shown in table 5.2 should be produced as part of this process.

Table 5.2: Documentation Requirements for Requirements Phase

Req	Document	I	II	III	IV	S	Remarks	Content Definition
R.0820	Software Requirements Specification	X	X	X	*	X	Content defined for all but Class IV software requirements can be included at system level at Class IV.	R.2650
R.0850	Requirements Trace	X	X	*	*	X	Bidirectional Trace for Class I, II, and S. Trace from parent for Class III, trace at system level at Class IV.	R.0850

5.4 Design

5.4.1 Design activities are performed after requirements are defined or modified. The objective of the design effort is to define software product(s) to be produced based on the defined requirements. Tasks normally performed during this phase include, but are not limited to: performing trade studies, doing build/buy analysis, evaluating software reuse, defining the software architecture, defining interfaces, and defining test planning/approaches. It is likely that the software design process will identify changes to/missing requirements. These updates/additions will need to be made to the appropriate requirements documentation, and the documentation be rebaselined. Changes to requirements will need to be evaluated to ensure that the design accounts for these changes. Upon completion of the design activities information is documented and reviewed prior to starting formal software development/procurement. The artifacts/products shown in table 5.3 should be produced as part of this process.

Table 5.3: Documentation Requirements for Design Phase

Req	Document	I	II	III	IV	S	Remarks	Content Definition
R.2680A	Interface Design Description	X	X	*		X	Content defined for Class I, II, S. Highly recommended for Class III.	R.2680A
R.2660A	Software Data Dictionary	X	X	*		X	Content defined for Class I, II, S. Recommended for Class III.	R.2660A
R.0900	Software Design Description	X	X	X	*	X	Content defined for all except Class IV. High level design description recommended for Class IV.	R.2670

Req	Document	I	II	III	IV	S	Remarks	Content Definition
R.0980	Software Test Plan	X	X	X	*	X	Content defined for all except Class IV. May be included at the system level for Class IV.	R.2620

5.5 Implementation

5.5.1 Implementation activities are performed after sufficient design detail has been completed. The objective of the implementation effort is to procure/produce software product(s) based on the agreed upon design. During this phase software coding/code modification is performed (as required). Coded software is normally loaded onto target platforms/emulators and configured. The resulting code is evaluated and modifications made. It is likely that the software development process will identify changes to the existing design. These updates/additions will need to be made to the appropriate design documentation, and the documentation be rebaselined. Changes to design will need to be evaluated to ensure that the as built system accounts for these changes. Upon completion of the development activities information is documented and reviewed prior to releasing a baseline. The artifacts/products shown in table 5.4 should be produced as part of this process.

Table 5.4: Documentation Requirements for Implementation Phase

Req	Document	I	II	III	IV	S	Remarks	Content Definition
R.2710A	Software User Manual	X	X	*		X	Content defined for Class I, II, S. Recommended for Class III systems employing a user interface.	R.2710A
R.0960	Software Version Description	X	X	X	X	X	Content defined for all classifications.	R.2720

5.6 Test

5.6.1 Test activities are performed upon completion/baseline of the software product and prior to the operational release of the software. The objective of the test effort is to show that software product(s) meet the goals/objectives/requirements specified. During this phase the baselined software configuration is verified and validated. The level of rigor associated with software testing is dependent upon the classification of the CSCI. Problems/discrepancies found during this phase need to be documented, analyzed, and dispositioned. Results of testing are reviewed and documented as well. It should be noted that the software test phase may not be complete until the systems/subsystems containing the CSCI(s) have been verified and validated as well. Software may be

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

released for operational use at the completion of this phase. The artifacts/products shown in table 5.5 should be produced as part of this process.

Table 5.5: Documentation Requirements for Test Phase

Req	Document	I	II	III	IV	S	Remarks	Content Definition
R.0980	Software Test Procedure(s)	X	X	X	*	X	Content defined for all but Class IV. Recommended for Class IV software testing. Can be included in system at Class IV.	R.2700A
R.0980	Software Test Report	X	X	X	*	X	Recommended for Class IV software testing. Can be included in system at Class IV.	R.2730A

5.7 Operations

5.7.1 The operations phase begins with the load of released operational software. Once the software is successfully loaded and verified the system/subsystem/software is cleared for use within its intended environment. Operations may continue for the duration of a project, test block, or experiment/measurement phase. Regardless of the duration, planning for the operations, maintenance, and retirement of the system/subsystem/software must be performed. Once this planning has been performed and documented the project needs to execute to the documented plan(s) for the duration of the Operations life cycle. The artifacts/products shown in table 5.6 should be produced as part of this process.

Table 5.6: Documentation Requirements for Operations Phase

Req	Document	I	II	III	IV	S	Remarks	Content Definition
R.1080	Software Operations Plan	X	X	X	*	X	Content needed for all but Class IV. Recommended for Class IV.	R.1080
R.0450	Software Maintenance Plan	X	X	X	*	X	Content defined for Class I, II, S. Recommended for Class III.	R.2630A
R.1080	Software Retirements Plan	X	X	X	*	X	Content needed for all but Class IV.	R.1080
R.2463	Flight Media Release	X	X	X	X	X	Form required for all classifications of flight software.	R.2463

5.8 Miscellaneous

5.8.1 There are several other processes that provide oversight/support functions during the software life cycle. These processes provide things like project and software assurance oversight as well as performing configuration management activities. The NPR 7150.2, NASA-STD-8719.13, and internal Center software requirements identify documentation/artifacts that need to be produced in support of these processes. The artifacts/products shown in table 5.7 should be produced/updated as part of the overall software engineering lifecycle.

Table 5.7: Documentation Requirements for Miscellaneous other activities

Req	Document	I	II	III	IV	S	Remarks	Content Definition
R.0470A	Software Cost Estimation	X	X	X		X	Estimation needed for all but Class IV. May be held at the project level.	R.0470A
R.3230A	Software Metrics Report	X	X			X	Content defined for Class I, II, S. May be held at the project level.	R.3230A
R.0480A	Software Schedule	X	X	X		X	Schedule needed for all but Class IV. May be held at the project level.	R.0480A
R.2690A	Software Change Request	X	X	X	*	X	Form required for all but Class IV. Strongly recommended for Class IV software.	R.2690A
R.2740A	Software Inspection Report	X	X	*	*	X	Form required for Class I, II, S. May be performed for Class III, IV at the request of SMA.	R.2740A
R.2740A	Software Peer Review Report	X	X	*	*	X	Form required for Class I, II, S. May be performed for Class III, IV at the request of SMA/the project.	R.2740A
R.2690A	Software Problem Report	X	X	*	*	X	Content defined for all but Class IV. Strongly recommended for Class IV software.	R.2690A

Appendix A: Definitions

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

Aviation safety. Safety efforts targeted at hazards associated with aviation activity. DPD-8700.1-001.

Commercial off-the-shelf software

1. Software defined by a market-driven need, commercially available, and whose fitness for use has been demonstrated by a broad spectrum of commercial users.
2. Software product available for purchase and use without the need to conduct development activities.
3. An item that a supplier offers to several acquirers for general use. ISO 24765:2010

Control flow. The sequence in which operations are performed during the execution of a computer program. ISO 24765:2010

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

Data flow. The sequence in which data transfer, use, and transformation are performed during the execution of a computer program. ISO 24765:2010

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-178

Deactivated code. Executable object code (or data) that is traceable to a requirement and, by design, is either (a) not intended to be executed (code) or used (data). RTCA DO-178

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-178

Embedded software. Software that is part of a larger system and performs some of the requirements of that system. ISO 24765:2010

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 24765:2010

Extraneous code. Executable code (or data) that is not traceable to any system or software requirement. RTCA DO-178

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. DPD-8700.1-001

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. DPR-7150.2-001.

Firmware. Combination of a hardware device and computer instructions or computer data that reside as read-only software on the hardware device. ISO 24765:2010

Government off-the-shelf software. Software supplied by the government for reuse in another project. ISO 24765:2010

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

Ground test safety. Ground safety efforts targeted at project-unique equipment. DPD-8700.1-001

Ground software. Software that could indirectly impact flight or test operations. This includes software supporting simulation, control room, data processing, or verification and validation test operations. DPR-7150.2-001-001.

Heritage software. Existing software that was produced/acquired before DPR-7150.2-001 was implemented. This software may have been classified in using DCP-S-007 Rev C or before.

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

Legacy software. See Heritage software.

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

Modified off-the-shelf software. 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 24765:2010

Modified condition/decision coverage. 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: (1) varying just that condition while holding fixed all other possible conditions or (2) varying just that condition while holding fixed all other possible conditions that could affect the outcome. RTCA DO-178

P(Center). Review of P(Center) is the short name for the requirement referred to as SWE-127 in NPR 7150.2.

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

Range safety. Safety efforts targeted at flight operations that threaten personnel and property to ensure that the risk of casualty/damage from an out-of-control impact is at or below an acceptable threshold. There is a recognized conceptual overlap with flight safety. It is generally recognized that aircrew are not included within the responsibility of range safety. DPD-8700.1-001

Reusable software product. A computer program used in the development, testing, analysis, or maintenance of a program or its documentation. ISO 24765:2010

Safety critical. 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. NPR 8715.3

Software. Computer programs, procedures, scripts, rules, and associated documentation and data pertaining to the development and operation of a computer system. NPR 7150.2

Software tool. 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. ISO 24765:2010

Statement coverage. Every statement in the program has been invoked at least once. RTCA DO-178

Structural coverage analysis. An evaluation of the code structure, including interfaces, exercised during requirements-based testing. RTCA DO-178

Subsystem. A secondary or subordinate system within a larger system. ISO 24765:2010

Support software. Software that aids in the development or maintenance of other software. ISO 24765:2010

System. The combination of elements that function together to produce the capability required to meet a need. NPR 7150.2

System software. Software designed to facilitate the operation and maintenance of a computer system and associated programs. ISO 24765:2010

Unit testing. A test of individual programs or modules in order to ensure that there are no analysis or programming errors. ISO 24765:2010

Validation. Proof that the product accomplishes the intended purpose. Validation may be determined by a combination of test, analysis, and demonstration. NPR 7123.1

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

Appendix B: Acronyms

AFRC -	Armstrong Flight Research Center
CIO	Chief Information Officer
CM	configuration management
CMMI	Capability Maturity Model Integration
COTS	commercial-off-the-shelf
CSCI	computer software configuration item
DCP	centerwide procedure
DO	RTCA document identification. Not an acronym.
DPD	policy directive
DPR	procedural requirements
FAR	Federal Acquisition Regulation
FMR	flight media release
GOTS	government-off-the-shelf
HDBK	handbook
IT	information technology
IV&V	independent verification and validation
MOA	Memorandum of Agreement
MOTS	modified-off-the-shelf
MOU	Memorandum of Understanding
NASA	National Aeronautics and Space Administration
NPD	NASA Policy Directive
NPR	NASA Procedural Requirements
OTS	off-the-shelf
PHA	preliminary hazard analysis
PLD	programmable logic device
RTCA	Radio Technical Commission for Aeronautics
SMA	Safety and Mission Assurance
SSSCA	software safety criticality assessment
STD	standard
SWE	software engineering
TA	technical authority

Before use, check the Master List to verify that this is the current version.

This document may be distributed outside of the Center.

Appendix C: Reference Documents

- a. NPR 2810.1, Security of Information Technology
- b. [DPD 8040.1-001](#), Configuration Management
- c. [DCP-P-025](#), Project Managers Manual
- d. [DCP-S-046](#), Flight Research Software Assurance Audit and Corrective Action Procedure
- e. [DCP-X-008](#), Tech Brief (T/B) and Mini Tech Brief (Mini T/B)
- f. [DCP-X-009](#), Airworthiness and Flight Safety Review Process
- g. [DCP-X-020](#), Flight Operational Readiness Review (ORR)
- h. [DCP-X-030](#), Dryden Center Management Council (DCMC) Reviews

Appendix D: Designated Governing Authority Allocations

ID	Center Director	Flight Operations	Research Engineering	Mission Systems	Mission Support	Acquisitions	Safety and Mission Assurance
R.0010							
R.0020							
R.0030							
R.0040							
R.0050							
R.0060	X						
R.0070	X						
R.0080	X						
R.0090							X
R.0100	X						
R.0110	X						
R.0120			X				
R.0130			X				
R.0140			X				
R.0150	X						
R.0160	X						
R.0170	X						
R.0181	X						
R.0190		X	X	X	X		X
R.0200		X	X	X	X		X
R.0210		X	X	X	X		X
R.0220							
R.0221							
R.0222							
R.0231							X
R.0232							X
R.0233							X
R.0241							X
R.0242							X
R.0261							X
R.0341							X
R.0342							X
R.0361							X
R.0371							X
R.0381							X
R.0450		X	X	X	X		X
R.0460		X	X	X	X		X
R.0470A		X	X	X	X		X

Before use, check the Master List to verify that this is the current version.
 This document may be distributed outside of the Center.

ID	Center Director	Flight Operations	Research Engineering	Mission Systems	Mission Support	Acquisitions	Safety and Mission Assurance
R.0480A		X	X	X	X		X
R.0490		X	X	X	X		X
R.0500		X	X	X	X		X
R.0510		X	X	X	X		X
R.0520		X	X	X	X		X
R.0530		X	X	X	X		X
R.0531							X
R.0532		X	X	X	X		X
R.0540A		X	X	X	X		X
R.0550		X	X	X	X		X
R.0560		X	X	X	X		X
R.0570							
R.0580		X	X	X	X		X
R.0590							
R.0600		X	X	X	X		X
R.0610		X	X	X	X		X
R.0681							X
R.0691							X
R.0692							X
R.0740		X	X	X	X		X
R.0750		X	X	X	X		X
R.0760		X	X	X	X		X
R.0770		X	X	X	X		X
R.0780		X	X	X	X		X
R.0790		X	X	X	X		X
R.0800		X	X	X	X		X
R.0810		X	X	X	X		X
R.0820		X	X	X	X		X
R.0830		X	X	X	X		X
R.0840A		X	X	X	X		X
R.0850		X	X	X	X		X
R.0860		X	X	X	X		X
R.0870		X	X	X	X		X
R.0880		X	X	X	X		X
R.0882		X	X	X	X		X
R.0890A		X	X	X	X		X
R.0900		X	X	X	X		X
R.0910		X	X	X	X		X
R.0920		X	X	X	X		X
R.0922		X	X	X	X		X

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

ID	Center Director	Flight Operations	Research Engineering	Mission Systems	Mission Support	Acquisitions	Safety and Mission Assurance
R.0924		X	X	X	X		X
R.0926A		X	X	X	X		X
R.0928		X	X	X	X		X
R.0930A		X	X	X	X		X
R.0940A		X	X	X	X		X
R.0950		X	X	X	X		X
R.0960		X	X	X	X		X
R.0970A		X	X	X	X		X
R.0980		X	X	X	X		X
R.0990		X	X	X	X		X
R.1000		X	X	X	X		X
R.1001		X	X	X	X		X
R.1002A		X	X	X	X		X
R.1004A		X	X	X	X		X
R.1006A		X	X	X	X		X
R.1008		X	X	X	X		X
R.1010		X	X	X	X		X
R.1020		X	X	X	X		X
R.1030		X	X	X	X		X
R.1040		X	X	X	X		X
R.1050		X	X	X	X		X
R.1060		X	X	X	X		X
R.1070		X	X	X	X		X
R.1080		X	X	X	X		X
R.1090		X	X	X	X		X
R.1100		X	X	X	X		X
R.1110		X	X	X	X		X
R.1120		X	X	X	X		X
R.1121		X	X	X	X		X
R.1131							X
R.1132							X
R.1171							X
R.1191							X
R.1211							X
R.1251							X
R.1261							X
R.1271							X
R.1291							X
R.1292							X
R.1321							X

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

ID	Center Director	Flight Operations	Research Engineering	Mission Systems	Mission Support	Acquisitions	Safety and Mission Assurance
R.1341							X
R.1401							X
R.1411							X
R.1421							X
R.1451							X
R.1461							X
R.1462							X
R.1501							X
R.1511							X
R.1531							X
R.1580		X	X	X	X		X
R.1620							X
R.1661							X
R.1731							X
R.1781							X
R.1811							X
R.1812							X
R.1813							X
R.1831							X
R.1851							X
R.1871							X
R.1881							X
R.1941							X
R.1942							X
R.2011							X
R.2041							X
R.2121							X
R.2131							X
R.2151							X
R.2221							X
R.2222							X
R.2231							X
R.2241							X
R.2331							X
R.2371							X
R.2391							X
R.2400		X	X	X	X		X
R.2410		X	X	X	X		X
R.2420A		X	X	X	X		X
R.2430		X	X	X	X		X

Before use, check the Master List to verify that this is the current version.
 This document may be distributed outside of the Center.

ID	Center Director	Flight Operations	Research Engineering	Mission Systems	Mission Support	Acquisitions	Safety and Mission Assurance
R.2440		X	X	X	X		X
R.2450A		X	X	X	X		X
R.2460		X	X	X	X		X
R.2461		X					
R.2462		X					
R.2463		X					
R.2464		X					
R.2465		X					
R.2466		X					
R.2491							X
R.2500							X
R.2511							X
R.2521							X
R.2600		X	X	X	X		X
R.2610		X	X	X	X		X
R.2620		X	X	X	X		X
R.2630A		X	X	X	X		X
R.2640A		X	X	X	X		X
R.2650		X	X	X	X		X
R.2660A		X	X	X	X		X
R.2670		X	X	X	X		X
R.2680A		X	X	X	X		X
R.2690A		X	X	X	X		X
R.2700A		X	X	X	X		X
R.2710A		X	X	X	X		X
R.2720		X	X	X	X		X
R.2730A		X	X	X	X		X
R.2740A		X	X	X	X		X
R.2750		X	X	X	X		X
R.2771							X
R.2781							X
R.2861		X	X	X	X		X
R.2990A						X	X
R.3000						X	X
R.3010						X	X
R.3020A						X	X
R.3030A						X	X
R.3040						X	X
R.3050						X	X
R.3060						X	X

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

ID	Center Director	Flight Operations	Research Engineering	Mission Systems	Mission Support	Acquisitions	Safety and Mission Assurance
R.3070A						X	X
R.3080						X	X
R.3101							X
R.3131							X
R.3132							X
R.3140A		X	X	X	X		X
R.3150		X	X	X	X		X
R.3160		X	X	X	X		X
R.3170		X	X	X	X		X
R.3180		X	X	X	X		X
R.3190							
R.3200							
R.3220		X	X	X	X		X
R.3230A		X	X	X	X		X

Before use, check the Master List to verify that this is the current version.
 This document may be distributed outside of the Center.

Appendix E: Requirements Mapping Matrix

ID	Source	Description	I	II	III	IV	S	Center
R.0010	SWE-002	Headquarters funds software engineering						X
R.0020	SWE-004	Headquarters benchmarks						X
R.0030	SWE-006	Headquarters maintains list of projects containing software						X
R.0040	SWE-098	Headquarters maintains process asset library						X
R.0050	SWE-129	Headquarters authorizes appraisals						X
R.0060	SWE-001	Effective date						X
R.0070	SWE-003	Center plan						X
R.0080	SWE-005	SW processes						X
R.0090	SWE-023	SW safety	X	X	X	X	X	
R.0100	SWE-099	Identify applicable practices						X
R.0110	SWE-100	Software engineering training						X
R.0120	SWE-101	Software training plan						X
R.0130	SWE-107	Center SW training plan						X
R.0140	SWE-108	Center SW engineering improve plan						X
R.0150	SWE-120	Alternate requirement request						X
R.0160	SWE-121	Document approved alternate requirements						X
R.0170	SWE-122	Non-IT and non-business						X
R.0181	SSS-007	Elevate disagreements with SSCA					X	X
R.0190	SWE-124	Direction for warrant authority						X

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

ID	Source	Description	I	II	III	IV	S	Center
R.0200	SWE-126	Considerations for waivers						X
R.0210	SWE-127	Review of "P(Center)"						X
R.0220	SWE-128	Compliance records						X
R.0221	SWE-139	Compliance with NPR requirements	X	X	X	X	X	
R.0222	SWE-140	Documenting Center tailoring of NPR requirements	X	X	X	X	X	
R.0231	SSS-002	Acquirer complies with software safety standard					X	
R.0232	SSS-003	Acquirer imposes software safety standard on provider					X	
R.0233	SSS-006	Acquirer SMA approves providers SSCA					X	X
R.0241	SSS-009	Provider SMA approves safety critical docs					X	
R.0242	SSS-010	Acquirer SMA approves docs with safety reqs					X	
R.0261	SSS-061	Provider collects safety objective evidence for acquirer acceptance					X	
R.0341	SSS-004	Provider adheres to software safety standard					X	
R.0342	SSS-062	Provider obtains acquirer and SMA approval for safety evidence of acceptance					X	X
R.0361	SSS-065	Provider addresses safe decommissioning of system in retirement plan					X	X
R.0371	SSS-066	Acquirer and provider SMA concurs on safe decommissioning of system in retirement plan					X	X
R.0381	SSS-017	Acquirer SMA maintains copy of waivers/deviations					X	X
R.0450	SWE-013	SW plan	X	X	X	X	X	
R.0460	SWE-014	Execute plan	X	X	X	X	X	
R.0470A	SWE-015	Cost estimation	X	X	X	X	X	
R.0480A	SWE-016	Schedule	X	X	X		X	

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

ID	Source	Description	I	II	III	IV	S	Center
R.0490	SWE-017	Training	X	X	X		X	
R.0500	SWE-018	Reviews	X	X	X		X	
R.0510	SWE-019	Life cycle	X	X	X		X	
R.0520	SWE-020	SW classification	X	X	X	X	X	
R.0530	SWE-021	SW classification changes	X	X	X	X	X	
R.0531	SWE-132	Independent classification assessment	X	X	X	X	X	
R.0532	SWE-133	Develop software safety criticality assessment	X	X	X	X	X	
R.0540A	SWE-022	SW assurance	X	X	X		X	
R.0550	SWE-024	Plan tracking	X	X	X		X	
R.0560	SWE-025	Corrective action	X	X	X		X	
R.0570	SWE-026	Changes	X	X	X	X	X	
R.0580	SWE-027	COTS, GOTS, MOTS	X	X	X		X	
R.0590	SWE-032	Intent of CMMI	X	X			X	X
R.0600	SWE-035	Supplier selection	X	X			X	
R.0610	SWE-038	Acquisition planning	X	X	X		X	
R.0681	SSS-020	Provider, with SMA, includes software in system safety analysis					X	
R.0691	SSS-031	Acquirer, with SMA, develops and maintains software safety plan					X	
R.0692	SSS-037	Provider includes list of Software safety plan contents					X	
R.0740	SWE-028	Verification planning	X	X	X		X	
R.0750	SWE-029	Validation planning	X	X	X		X	

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

ID	Source	Description	I	II	III	IV	S	Center
R.0760	SWE-030	Verification results	X	X	X	X	X	
R.0770	SWE-031	Validation results	X	X	X	X	X	
R.0780	SWE-033	Options for Acquisition	X	X	X		X	
R.0790	SWE-034	Acceptance Criteria	X	X	X		X	
R.0800	SWE-036	SW processes and tasks	X	X	X		X	
R.0810	SWE-037	Milestones	X	X	X		X	
R.0820	SWE-049	Documented Requirements	X	X	X	X	X	
R.0830	SWE-050	SW requirements	X	X	X		X	
R.0840A	SWE-051	Flow-down and derived req.	X	X			X	
R.0850	SWE-052	Bi-directional trace	X	X	X		X	
R.0860	SWE-053	Manage req. change	X	X	X	X	X	
R.0870	SWE-054	Corrective Action	X	X	X		X	
R.0880	SWE-055	Requirements Validation	X	X	X		X	
R.0882	Center Unique Ref DO-178	High Level Algorithm Review	X	X	X			
R.0890A	SWE-059	Bi-directional trace	X	X			X	
R.0900	SWE-056	Document design	X	X	X		X	
R.0910	SWE-057	Architecture	X	X	X		X	
R.0920	SWE-058	Detailed design	X	X	X		X	
R.0922	Center Unique Ref DO-178	Low Level Algorithm Review	X	X				
R.0924	Center Unique Ref DO-178	Software Architecture Review vs. high level requirements	X	X				

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

ID	Source	Description	I	II	III	IV	S	Center
R.0926A	Center Unique Ref DO-178	Consistency	X	X				
R.0928	Center Unique Ref DO-178	Review of software architecture partitioning	X	X	X			
R.0930A	SWE-064	Maintain traceability	X	X			X	
R.0940A	SWE-061	Coding standards	X	X			X	
R.0950	SWE-062	Unit test	X	X	X		X	
R.0960	SWE-063	Version description	X	X	X	X	X	
R.0970A	SWE-070	Models, simulations, tools	X	X			X	
R.0980	SWE-065	Plan, procedures, reports	X	X	X		X	
R.0990	SWE-066	Perform testing	X	X	X	X	X	
R.1000	SWE-067	Test for compliance	X	X	X		X	
R.1001	SWE-135	Static analysis tools	X	X			X	
R.1002A	Center Unique Ref DO-178	Perform structural coverage analysis	X	X	X			
R.1004A	Center Unique Ref DO-178	Utilize structural coverage analysis to identify and perform additional testing requirements	X	X	X			
R.1006A	Center Unique Ref DO-178	Extraneous/dead code identification and removal	X	X	X			
R.1008	Center Unique Ref DO-178	Deactivated code analysis	X	X	X			
R.1010	SWE-068	Evaluate test results	X	X	X		X	
R.1020	SWE-069	Document defect and track	X	X	X	X	X	
R.1030	SWE-071	Update plans and procedures	X	X	X		X	
R.1040	SWE-072	Maintain traceability	X	X	X		X	

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

ID	Source	Description	I	II	III	IV	S	Center
R.1050	SWE-073	Platform or Hi-Fidelity simulation.	X	X	X		X	
R.1060	SWE-074	Document maintenance plans	X	X	X		X	
R.1070	SWE-075	Plan operations, maintenance, and retirement	X	X	X		X	
R.1080	SWE-076	Implement plans	X	X	X		X	
R.1090	SWE-077	Deliver software product	X	X	X	X	X	
R.1100	SWE-078	As-built documentation	X	X	X		X	
R.1110	SWE-087	Requirements and test plans	X	X	X		X	
R.1120	SWE-088	Checklist, criteria, and tracking	X	X	X		X	
R.1121	SWE-137	Reporting results	X	X			X	
R.1131	SSS-001	Acquirer SMA performs SSCA Part 1					X	
R.1132	SSS-005	Provider SMA performs SSCA					X	
R.1171	SSS-008	Provider maintains SSCA					X	
R.1191	SSS-035	Provider, with SMA, develops, baselines, and configuration manages software safety plan per contract/MOU/MOA					X	
R.1211	SSS-019	Provider, with SMA, performs safety analysis on OTS and reused software					X	
R.1251	SSS-015	Prepare waiver/deviation package if any requirements cannot be met					X	
R.1261	SSS-016	SMA evaluates and submits waiver/deviation package					X	
R.1271	SWE-134	Software safety modes and states	X	X			X	
R.1291	SSS-038	Provider identifies/tags for software safety requirements					X	
R.1292	SSS-041	Provider SMA concurs on software safety requirements identification/tagging					X	
R.1321	SSS-039	Provider includes hardware/software/operator safety constraints in requirements document					X	

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

ID	Source	Description	I	II	III	IV	S	Center
R.1341	SSS-043	Provider unique tags software design implementing safety features/methods					X	
R.1401	SSS-054	Provider verifies software safety requirements by test					X	
R.1411	SSS-055	Provider performs testing verifying hazards are eliminated/controlled					X	
R.1421	SSS-056	Provider includes unit/component testing for safety aspects of software					X	
R.1451	SSS-053	Provider SMA concurs software safety verification results					X	
R.1461	SSS-051	Provider SMA concurs on test plans/procedures					X	
R.1462	SSS-052	Provider includes software conditions impacting performance in system testing					X	
R.1501	SSS-057	Provider verifies/validates by tests including loads, stress, and off-nominal conditions					X	
R.1511	SSS-058	Provider verifies by tests including correct and safe operations, transitioning to safe state					X	
R.1531	SSS-059	Provider, with SMA, verifies by analysis, inspection, or demonstration if cannot by test					X	
R.1580	SWE-060	Design→code	X	X	X		X	
R.1620	SWE-086	Continuous risk management	X	X	X		X	
R.1661	SSS-023	Provider identifies new or updated software safety requirements					X	
R.1731	SSS-033	Acquirer SMA verifies safety contents in contract/MOU/MOA					X	
R.1781	SSS-021	Acquirer assigns SMA as approving member					X	
R.1811	SSS-034	Acquirer SMA evaluates and approves provider SW safety plan					X	
R.1812	SWE-130	Develop software safety plan					X	
R.1813	SWE-138	Generate plan					X	
R.1831	SSS-024	Provider SMA re-evaluates SSCA, obtains acquirer SMA approval for changes					X	
R.1851	SSS-011	Provider traces software safety requirements to system hazards					X	

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

ID	Source	Description	I	II	III	IV	S	Center
R.1871	SSS-013	Provider SMA verifies Configuration management implementation					X	
R.1881	SSS-012	Provider SMA concurs with trace results					X	
R.1941	SSS-022	Provider develops and maintains evidence of SSS compliance					X	
R.1942	SSS-040	Provider SMA analyzes the software safety requirements per criteria					X	
R.2011	SSS-042	Provider updates software requirements analysis results in system safety data package.					X	
R.2041	SSS-044	Provider SMA performs software design analysis per criteria					X	
R.2121	SSS-046	Provider SMA concurs on design hazards/controls and updated data products					X	
R.2131	SSS-045	Provider updates system safety data package based on software design analysis results					X	
R.2151	SSS-045	Provider, with SMA, performs code analysis on safety critical code					X	
R.2221	SSS-047	Provider identifies/tags of safety critical code					X	
R.2222	SSS-050	Provider SMA concurs on code hazards/controls and updated data products					X	
R.2231	SSS-049	Provider updates system safety data products based on code analysis results					X	
R.2241	SSS-014	Provider gives access to acquirer and acquirer SMA					X	
R.2331	SSS-060	Provider SMA verifies any new hazards updated in system safety data package					X	
R.2371	SSS-063	Provider SMA verifies regression testing required for implementing new requirements/changes	X	X	X	X	X	
R.2391	SSS-064	Provider SMA evaluates user manuals and procedures for safe operation/impacts	X	X	X	X	X	
R.2400	SWE-079	Develop CM plan	X	X	X	X	X	
R.2410	SWE-080	Track and evaluate changes	X	X	X	X	X	
R.2420A	SWE-081	Identify S/W configuration items	X	X	X	X	X	
R.2430	SWE-082	Authorize changes	X	X	X	X	X	

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

ID	Source	Description	I	II	III	IV	S	Center
R.2440	SWE-083	Maintain records	X	X	X	X	X	
R.2450A	SWE-084	Perform configuration audits	X	X			X	
R.2460	SWE-085	Implement procedures	X	X	X	X	X	
R.2461	AFRC Unique (flight software)	Labeling of software media	X	X	X	X		
R.2462	AFRC Unique (flight software)	Version description document	X	X	X	X		
R.2463	AFRC Unique (flight software)	Flight media release form included in version description document	X	X	X	X		
R.2464	AFRC Unique (flight software)	Flight software installation procedure	X	X	X	X		
R.2465	AFRC Unique (flight software)	Specification of flight(s) on flight media release form	X	X	X	X		
R.2466	AFRC Unique (flight software)	Confirmation of correct software version before flight	X	X	X	X		
R.2491	SSS-029	Provider evaluate discrepancy reports for safety impacts					X	
R.2501	SSS-028	Provider SMA concurs on all software changes					X	
R.2511	SSS-030	Provider traces discrepancies/problems/failures to system hazards					X	
R.2521	SSS-027	Provider SMA analyzes software changes for safety impacts					X	
R.2600	SWE-102	SW development/mgt. plan	X	X	X		X	
R.2610	SWE-103	SW configuration mgt. plan	X	X	X		X	
R.2620	SWE-104	SW test plan	X	X	X		X	

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

ID	Source	Description	I	II	III	IV	S	Center
R.2630A	SWE-105	SW maintenance plan	X	X			X	
R.2640A	SWE-106	SW assurance plan	X	X			X	
R.2650	SWE-109	SW requirements spec	X	X	X		X	
R.2660A	SWE-110	SW data dictionary	X	X			X	
R.2670	SWE-111	SW design description	X	X	X		X	
R.2680A	SWE-112	Interface design description	X	X			X	
R.2690A	SWE-113	SW change request/problem report	X	X	X	X	X	
R.2700A	SWE-114	SW test procedures	X	X			X	
R.2710A	SWE-115	SW user's manual	X	X			X	
R.2720	SWE-116	SW version description	X	X	X	X	X	
R.2730A	SWE-118	SW test report	X	X			X	
R.2740A	SWE-119	SW inspection/peer review	X	X			X	
R.2750	SWE-125	Compliance matrix	X	X	X	X	X	
R.2771	SWE-136	Tool accreditation	X	X	X		X	
R.2781	SSS-018	Provider determines if tools impact safety	X	X	X		X	
R.2861	SWE-131	Develop IV&V execution plan as needed	X	X			X	
R.2990A	SWE-042	Source code access	X	X			X	
R.3000	SWE-044	SW measurement data	X	X	X		X	
R.3010	SWE-039	Insight into test	X	X	X		X	
R.3020A	SWE-040	Electronic access	X	X			X	

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

ID	Source	Description	I	II	III	IV	S	Center
R.3030A	SWE-041	Open source	X	X			X	
R.3040	SWE-043	Track change request	X	X	X		X	
R.3050	SWE-045	Joint audits	X	X	X		X	
R.3060	SWE-046	SW schedule	X	X	X		X	
R.3070A	SWE-047	Traceability data	X	X			X	
R.3080	SWE-048	Solicitation	X	X	X		X	
R.3101	SSS-032	Acquirer identifies additional software safety requirements, includes them in contract/MOU/MOA						
R.3131	SSS-025	Provider assigns SMA as voting member of provider software change control process						
R.3132	SSS-026	Acquirer assigns SMA as voting member of acquirer software change control process						
R.3140A	SWE-089	Basic measurements	X	X			X	
R.3150	SWE-091	SW measurement areas	X	X	X		X	
R.3160	SWE-092	Collection and storage	X	X	X		X	
R.3170	SWE-093	Analyze data	X	X			X	
R.3180	SWE-094	Report analysis	X	X			X	
R.3190	SWE-095	Mission directorates define software measurement system						X
R.3200	SWE-096	Mission directorates define software measurement objectives						X
R.3220	SWE-090	Objectives	X	X			X	
R.3230A	SWE-117	SW metrics report	X	X			X	

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

Appendix F: Compliance/Reference Information

F.1 NPR 7150.2 to DPR-7150.2-001 software classification reference

F.1.1 A cursory review of the of the classification methodology defined in Chapter 3 of the DPR would leave one to believe that there is little correlation between it and the software classifications found in Appendix E of NPR 7150.2. Correlations do exist between the NPR and DPR classification methods when they are applied to the traditional Center based applications. This section provides the background information used to draw these correlations.

F.1.1.1 The criteria called out for identifying/determining if software is safety critical does not differ from the requirement identified in NPR 7150.3, SWE-133. This requirement states: The project, in conjunction with the Safety and Mission Assurance organization shall determine the software safety criticality in accordance with NASA-STD-8739.8. Appendix A of this standard defines the litmus test as follows:

a. Resides in a safety-critical system (as determined by hazard analysis AND at least one of the following apply:

- (1) Causes or contributes to a hazard.
- (2) Provides control or mitigation for hazards.
- (3) Processes safety-critical commands or data.
- (4) Detects and reports, or takes corrective action, if the system reaches.

b. Specific hazardous state.

- (1) Mitigates damage if a hazard occurs.
- (2) Resides on the same system (processor) as safety-critical software.

c. Processes data or analyzes trends that lead directly to safety decisions.

d. Provides full or partial verification or validation of safety-critical systems, including hardware or software subsystems.

The classification methodology called out in Section 3.3 of the DPR makes the equivalent call outs for determining if software is safety critical. Once this determination is made, all safety critical requirements called out in NPR 7150.2, Appendix D, are evoked. In addition, the requirements specified in NASA-STD-8719.13 are evoked as well.

F.1.1.2 The requirements associated with DPR-7150.2-001 Class I and II software were selected so they would correlate with the requirements called out for in NPR 7150.2 Class C software. Class III software requirements meet or exceed the Class D requirements. Class IV software exceeds the requirements found in Class E. The majority system/software projects involve either airborne platforms, or engineering/research facility applications. In addition, the vast majority of the airborne platforms/projects at the Center do not fall into the category of large scale aeronautics vehicle (Greater than \$250M life cycle cost). As a result, the NPR 7150.2 classifications that are likely to be implemented will fall into the definitions found in Class C-E. As a result, this DPR is tailored to implement the requirement called out for Class C-E software.

Implementing the hazard based classification methodology provides the means to automatically account for the airborne vehicle system litmus test taken from RTCA DO-178. The NPR makes the following call out: Software whose anomalous behavior would cause or contribute to a failure of system function resulting in a minor failure condition for the airborne vehicle. DO-178, section 2.1.2. The system safety assessment process determines the impact of the software design and implementation on system safety using information provided by the software life cycle processes. Therefore, the determination of a minor failure condition is derived from a hazard based assessment. In the case of airborne research vehicles, the failure to gather noncritical project data for a given sortie constitutes a loss of sortie/mission and return to base. This constitutes a minor failure in terms per NPR 7150.2. As a result, it is Class D software.

F.1.1.2 Provides the following definition of a minor failure condition: "Failure conditions that would not significantly reduce aircraft safety, and that would involve crew actions that are well within their capabilities." According to DO-178, section 2.1.2, the system safety assessment process determines the impact of the software design and implementation on system safety using information provided by the software life cycle processes. Therefore, the determination of a minor failure condition is derived from a hazard based assessment. In the case of airborne research vehicles, the failure to gather noncritical project data for a given sortie constitutes a loss of sortie/mission and return to base. This constitutes a minor failure in terms per NPR 7150.2. As a result, it is Class D software.

Class IV software provides relief to the Class III airborne applications in cases where the failure of the software results in nothing more than a minor productivity impact to a small number of users. Examples of this are noncritical real time monitoring functions and data processing algorithms. In these cases, applying the full suite of Class III software requirements may be excessive. As a result, some requirements have been relaxed in the DPR. While Class IV software exceeds the NPR Class E requirements, there are no provisions allowing for Class E software to be implemented on an airborne platform. According to the NPR Class D software includes: "2. Software whose anomalous behavior would cause or contribute to a failure of as system function with no effect on airborne vehicle operational capability or pilot workload." This definition is

Before use, check the Master List to verify that this is the current version.

This document may be distributed outside of the Center.

derived from DO-178B, Level E. In this case, no effect (or level E) is defined as “Failure conditions which do not affect the operational capability of the aircraft or increase crew workload.” Section 2.2.2 of DO-178 provides the following additional information related to how Level E software is handled. “Once software has been confirmed as level E by the certification authority, no further guidelines of this document (DO-178) apply.” This implies that NPR 7150.2 levies a more stringent set of requirements for airborne software that has no effect on the platform than those using DO-178 as the guidance for certification. DPR-7150.2-001 has taken a measured approach when dealing with the instances where Class IV software is used in airborne applications. To begin with, the P(Center) requirements defined at Class III may, or may not be applied at Class IV. In addition, six of the software engineering life cycle requirements have been delegated to the system level. The following table contains a complete list of differences between Class III and Class IV flight software.

Req	SWE Requirement	III	IV	Remarks
R.0480A	SWE-016	X		P (Center) software schedule not required for Class IV.
R.0500	SWE-018	X		Reviews may be performed at the system level for Class IV.
R.0510	SWE-019	X		P (Center) software life cycle not required for Class IV.
R.0540A	SWE-022	X		Software assurance activities may be limited to verification of flight software loads for Class IV.
R.0740	SWE-028	X		P (Center) software verification planning not required for Class IV.
R.0750	SWE-029	X		P (Center) software validation planning not required for Class IV.
R.0780	SWE-033	X		Level of detail related to software acquisition assessment (build/buy) left to the discretion of the project for Class IV.
R.0790	SWE-034	X		Software acceptance criteria delegated to the system level for Class IV.
R.0800	SWE-036	X		P (Center) software processes and tasks not required for Class IV.
R.0810	SWE-037	X		P (Center) software milestones not required for Class IV.
R.0610	SWE-038	X		Level of documentation related to software acquisition decisions left to the discretion of the project for Class IV.

Before use, check the Master List to verify that this is the current version.

This document may be distributed outside of the Center.

Req	SWE Requirement	III	IV	Remarks
R.3010	SWE-039	X		P (Center) software insight into test activities not required for Class IV.
R.3000	SWE-044	X		P (Center) software measurement data not required for Class IV.
R.3060	SWE-046	X		Level of insight into supplier software schedule left to the discretion of the project for Class IV.
R.3080	SWE-044	X		P (Center) software solicitation data not required for Class IV.
R.0830	SWE-050	X		Software requirements may be defined at the system level for Class IV.
R.3080	SWE-057	X		P (Center) software architecture definition not required for Class IV.
R.0830	SWE-060	X		Software design→code may be captured informally or at a high level for Class IV.
R.0950	SWE-062	X		P (Center) software unit level tests not required for Class IV.
R.0980	SWE-065	X		P (Center) software plans, procedures, and reports not required for Class IV.
R.1010	SWE-068	X		P (Center) Evaluation of the performance of software may be evaluated at the system level for Class IV.
R.1060	SWE-072	X		Requirements traceability needs to be maintained to the level of decomposition for Class IV. Note: This may stop at the system level.
R.2600	SWE-102	X		P (Center) compliance to content defined in SWE is not required for Class IV.
R.2610	SWE-103	X		P (Center) compliance to content defined in SWE is not required for Class IV.
R.2620	SWE-104	X		P (Center) compliance to content defined in SWE is not required for Class IV.
R.2640	SWE-109	X		P (Center) compliance to content defined in SWE is not required for Class IV.
R.2670	SWE-111	X		P (Center) compliance to content defined in SWE is not required for Class IV.

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

Req	SWE Requirement	III	IV	Remarks
R.2771	SWE-136	X		For Class IV software may be accredited at the system level. It is unlikely that Note 5 in appendix D of NPR 7150.2 will apply to Class IV CSCIs.

F.2 Application of NPR 7150.2 to Certified Airborne Platforms

F.2.1 The vast majority of the airborne platforms in use at the Center are existing, legacy airframes. In almost every case, the software contained within the aircraft avionics systems have been developed, integrated, and tested using defined system/software development processes. These vehicles are typically maintained in accordance with published manufacturers processes and procedures with the approval of the appropriate certification authority. Typically, these packages are reviewed, tested, and approved by the appropriate certification authority. These maintenance actions, including software upgrades, may be performed with minimal review. The Center belief is that this approach is in line with the following statement in NPR 7150.2: "This NPR does not supersede more stringent requirements imposed by individual organizations and other Federal Government agencies." Since the failure to comply with airworthiness directives made by the applicable certification authority could result in an aircraft being grounded compliance to these directives becomes mandatory. The Center will perform the work, document the configuration, and perform the necessary regression testing in accordance with the instructions provided by the manufacturer.

F.2.1.1 In most cases, software will be classified in accordance with the appropriate industry standard. These classification categories will differ from those found in NPR 7150.2. The Center will not attempt to perform an independent classification assessment or gap analysis for certified production aircraft software being maintained in accordance with direction from the airframe manufacturer/certification authority.

F.2.2 When production software is included as part of a research system/experiment then the production software will be captured as a research CSCI. At that time, the software will be reclassified in accordance with this DPR and managed accordingly. This process will remain in effect until the system/software is restored to its production configuration.

F.2.3 When legacy research systems/software is to be reused as part of a new project then the software will be captured as a new CSCI and treated as a MOTS. In no case will this type of software be treated as production software.

F.3 Application of DPR-7150.2-001 to business and information technology infrastructure software.

F.3.1 While the application of this DPR to business and IT infrastructure software is possible it may not be the best fit. In addition, use of the typical tools for performing

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

hazard assessments (i.e., [DCP-S-002](#)) have been tailored to support typical Center aerospace projects rather than IT infrastructure projects. The software classification criteria defined in Chapter 3 should be cross referenced with the software classifications found in Appendix E of NPR 7150.2 to ensure that gaps do not exist between the NPR expectations for software classifications and those found in the DPR.

F.3.1.1 Software classified as Class F in NPR 7150.2 closely match the Class I and II, requirements found in the DPR with the following exceptions.

Req	SWE Requirement	Remarks
R.0170	SWE-122	Delegation of TA for software other than business or IT infrastructure not required for Class F per NPR.
R.0590	SWE-032	Intent/implementation of CMMI not required for Class F per NPR.
R.0882	Center Unique	High level algorithm review not required for Class F software per the NPR.
R.0922	Center Unique	Low level algorithm review not required for Class F software per the NPR.
R.0924	Center Unique	Review of relationship between software architecture vs. high level algorithms not required for Class F software per the NPR.
R.0926A	Center Unique	Review of relationship between software components not required for Class F software per the NPR.
R.0928	Center Unique	Review of software partitioning not required for Class F software per the NPR.
R.1001	SWE-135	Use of static analysis tools not required for Class F software per the NPR.
R.1002A	Center Unique	Structural coverage analysis not required for Class F software per the NPR.
R.1004A	Center Unique	Utilize results of structural coverage analysis to identify additional tests not required for Class F software per the NPR.
R.1006A	Center Unique	Identify and remove extraneous/dead code not required for Class F software per the NPR.
R.1008	Center Unique	Identification of deactivated code not required for Class F software per the NPR.
R.2461	Center Unique	Flight software media labeling requirements are not applicable.
R.2463	Center Unique	Flight software load procedure not applicable.
R.2464	Center Unique	Flight software installation procedure not applicable.
R.2465	Center Unique	Specification of applicable flights on FMR not applicable.
R.2466	Center Unique	Confirmation of correct software version before flight not applicable.
R.2771	SWE-136	Tool accreditation not required for Class F software per the NPR.

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

F.3.1.2 Software classified as Class G in NPR 7150.2 closely match the Class III, requirements found in the DPR with the following exceptions.

Req	SWE Requirement	Remarks
R.0170	SWE-122	Delegation of TA for software other than business or IT infrastructure not required for Class G per NPR.
R.0600	SWE-035	Supplier select required for Class G per NPR not required in Class III.
R.0840A	SWE-051	Flow down of derived requirements required for Class G per NPR not required in Class III.
R.0882	Center Unique	High level algorithm review not required for Class G software per the NPR.
R.0890A	SWE-059	Bi-directional trace of requirements required for Class G per NPR not required in Class III.
R.0922	Center Unique	Low level algorithm review not required for Class G software per the NPR.
R.0924	Center Unique	Review of relationship between software architecture vs high level algorithms not required for Class G software per the NPR.
R.0926A	Center Unique	Review of relationship between software components not required for Class G software per the NPR.
R.0928	Center Unique	Review of software partitioning not required for Class G software per the NPR.
R.0930A	SWE-064	Main Traceability required for Class G per NPR not required in Class III.
R.0940	SWE-061	Coding standards required for Class G per NPR not required in Class III.
R.1002A	Center Unique	Structural coverage analysis not required for Class G software per the NPR.
R.1004A	Center Unique	Utilize results of structural coverage analysis to identify additional tests not required for Class G software per the NPR.
R.1006A	Center Unique	Identify and remove extraneous/dead code not required for Class G software per the NPR.
R.1008	Center Unique	Identification of deactivated code not required for Class G software per the NPR.
R.1121A	SWE-115	Results Reporting required for Class G per NPR not required in Class III.
R.2450	SWE-084	Configuration Audits required for Class G per NPR not required in Class III.
R.2461	Center Unique	Flight software media labeling requirements are not applicable.
R.2463	Center Unique	Flight software load procedure not applicable.

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

Req	SWE Requirement	Remarks
R.2464	Center Unique	Flight software installation procedure not applicable.
R.2465	Center Unique	Specification of applicable flights on FMR not applicable.
R.2466	Center Unique	Confirmation of correct software version before flight not applicable.
R.2630A	SWE-105	Software maintenance plan required for Class G per NPR not required in Class III.
R.2640A	SWE-106	Software assurance plan required for Class G per NPR not required in Class III.
R.2660A	SWE-110	Software data dictionary required for Class F per NPR not required in Class III.
R.2680A	SWE-112	Software interface design description required for Class G per NPR not required in Class III.
R.2700A	SWE-114	Software test procedures required for Class G per NPR not required in Class III.
R.2710A	SWE-115	Software user manual required for Class G per NPR not required in Class III.
R.2730A	SWE-118	Software test report required for Class G per NPR not required in Class III.
R.2740A	SWE-119	Software inspection/peer review report required for Class G per NPR not required in Class III.
R.2771	SWE-136	Tool accreditation not required for Class G software per the NPR.
R.2990A	SWE-042	Access to source code required for Class G per NPR not required in Class III.
R.3020A	SWE-040	Electronic access to supplier artifacts required for Class G per NPR not required in Class III.
R.3030A	SWE-041	Open Source disclosures required for Class G per NPR not required in Class III.
R.3070A	SWE-047	Traceability data required for Class G per NPR not required in Class III.
R.3140A	SWE-089	Maintenance of metrics data required for Class G per NPR not required in Class III.
R.3170	SWE-093	Analysis of metrics data required for Class G per NPR not required in Class III.

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

F.3.1.3 Software classified as Class H in NPR 7150.2 closely match the Class IV, requirements found in the DPR with the following exceptions.

Req	SWE Requirement	Remarks
R.3180	SWE-094	Report results of analysis of metrics data required for Class G per NPR not required in Class III.
R.3220	SWE-090	Document metrics objective required for Class G per NPR not required in Class III.
R.3230A	SWE-117	Software metrics report required for Class G per NPR not required in Class III.
R.0170	SWE-122	Delegation of TA for software other than Business or IT infrastructure not required for Class H per NPR.
R.0450	SWE-013	Software plans not required for Class H software per the NPR.
R.0460	SWE-014	Execute software plans not required for Class H software per the NPR.
R.0470A	SWE-015	Perform cost estimation not required for Class H software per the NPR.
R.0760	SWE-030	Verification results not required for Class H software per the NPR.
R.0770	SWE-031	Validation results not required for Class H software per the NPR.
R.0790	SWE-035	Supplier select required for Class H per NPR not required in Class IV.
R.0860	SWE-053	Manage requirement changes not required for Class H software per the NPR.
R.0960	SWE-063	Manage requirement changes not required for Class H software per the NPR.
R.0990	SWE-066	Perform testing not required for Class H software per the NPR.
R.1090	SWE-077	Deliver software product not required for Class H software per the NPR.
R.2400	SWE-079	Develop configuration management plan not required for Class H software per the NPR.
R.2410	SWE-080	Track and evaluate changes not required for Class H software per the NPR.

Before use, check the Master List to verify that this is the current version.
This document may be distributed outside of the Center.

F.3.1.4 As has been previously stated, software acquisition/ development activities falling into the business and IT infrastructure software classifications (Class F-H per NPR 7150.2) may elect to follow the requirements found in NPR 7150.2 in lieu of the requirements found in this DPR. Projects may elect to do this without the need to generate a waiver to this DPR.

Req	SWE Requirement	Remarks
R.2461	Center Unique	Flight software media labeling requirements are not applicable.
R.2462	Center Unique	Generate version description document for flight software requirements are not applicable.
R.2463	Center Unique	Flight software load procedure not applicable.
R.2464	Center Unique	Flight software installation procedure not applicable.
R.2465	Center Unique	Specification of applicable flights on FMR not applicable.
R.2466	Center Unique	Confirmation of correct software version before flight not applicable.
R.2720	SWE-116	Develop version description document not required for Class H software per the NPR.

F.4 Center approach to meeting the intent of SWE-032 (CMMI)

F.4.1 NPR 7150.2 SWE-032 identifies a requirement to acquire, develop, and maintain software in compliance with the CMMI process module. The expectations regarding the level of compliance vary by the software classification specified in NPR 7150.2. The full requirement and associated expectations can be found in section 2.5.1 of the NPR.

F.4.2 This section of the DPR provides the Center's approach to partially complying with the CMMI Maturity Level 2 model for NPR 7150.2 Class C, and Class C-E Safety Critical software. The software classification methodology, and requirements called out in this DPR are based on the assumption that software acquisition and development projects being managed at the Center fall into the category of aeronautics vehicles with a life cycle cost that is less than \$250m. If this is the case, then SWE-032 becomes a P(Center) requirement. According to the NPR, "The required CMMI-DEV Maturity Level for Class C software will be defined per Center or project requirements." This means the Center may define their approach to showing compliance to CMMI Level 2.

F.4.3 The Center has a number of defined/implemented processes that define the processes that projects are expected to follow. By adhering to these processes, projects will achieve the organizational/process maturity needed to show partial

compliance with the CMMI level 2 model as specified in the NPR. Projects needing to show compliance with SWE-032 (R.0590) must ensure that the following DPDs, DPRs, and DCPs are implemented.

a) Project Planning

(1) DPD-1000.1.001, Governance and Strategic Management Handbook

(2) DPR-7123.1-001, Systems Engineering Requirements

(3) DPR-7150.2-001, Software Engineering Requirements

(4) DCP-P-025, Project Managers Manual

a) Project Monitoring and Control

(1) DPD-1000.1-001, Governance and Strategic Management Handbook

(2) DCP-X-008, Tech Brief and Mini Tech Brief

(3) DCP-X-009, Airworthiness and Flight Safety Review Process

(4) DCP-X-020, Flight Operational Readiness Review

(5) DCP-X-030, Dryden Center Management Council Reviews

b) Requirements Management

(1) DPR-7123.1-001, Systems Engineering Requirements

(2) DPR-7150.2-001, Software Engineering Requirements

(3) DCP-P-025, Project Managers Manual

a) Configuration Management

(1) DPD-8040.1-001, Configuration Management

(2) DPR-7150.2-001, Software Engineering Requirements

(3) DCP-O-002, Aircraft Work Order Procedure

(4) DCP-O-030, Aircraft Documentation

(5) DCP-P-025, Project Managers Manual

(6) DPR17123.2-001, Waivers and deviations to Technical Requirements and Standards

a) Process and product quality assurance

(1) DCP-S-007, Software Assurance

(2) DCP-S-046, Flight Research Software Assurance Audit and Corrective Action Procedure

F.5 Center approach to meeting the intent of SWE-003 and SWE-108 (Center SW Engineering Improvement Plan)

F.5.1 NPR 7140.2 SWE-003 identifies a requirement to maintain, staff, and implement a plan to continually advance the Center's in-house software engineering capability and monitor the software engineering capability of NASA's contractors. Furthermore, SWE-108 identifies specific information required to be included in the plan. The full requirements and associated expectations can be found in sections 2.1.3 and 5.1.7 of the NPR.

F.5.2 This section of the DPR provides the Center's approach to complying with the intent of the Software Engineering Improvement plan.

F.5.3 The intent of the software engineering improvement plan is to provide a basis for assessment of the center's process improvement by OCE. While the center does not have a formally documented plan, the center software technical authority maintains a list of candidate improvements and coordinates efforts and monitors progress toward making those improvements.

Document History Log

Review Date: 05/13/14

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 SMA 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

Before use, check the Master List to verify that this is the current version.

This document may be distributed outside of the Center.