DRYDEN PROCEDURAL REQUIREMENT

Directive:	**DPR-7150.2-001, Baseline-1**
Effective Date:	June 3, 2010
Expiration Date:	June 3, 2015

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

SUBJECT: Dryden Software Engineering Requirements

RESPONSIBLE OFFICE: X/Office of the Chief Engineer

TABLE OF CONTENTS

DISTRIBUTION: Dryden Document Library
If distribution is intended for NASA only, include a statement to identify the restrictions for release.

THIS PAGE INTENTIONALLY LEFT BLANK

PREFACE

P.1 Purpose

a. This procedure brings the Dryden Flight Research Center into compliance with NASA Policy Directive NPD 2820.1 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 process, but instead provides a single set of requirements for all center software development activities. This will allow organizations at Dryden 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 system defined in NPR 7150.2 to reduce confusion and improve traceability to other common aeronautics standards and existing Dryden processes, including RTCA DO-178 and DCP-S-002.

P.2 Applicability

a. This DPR is applicable to Dryden and other NASA employees visiting, detailed, or assigned to DFRC 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 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 2820.1, NASA Software Policies

b. NPR 7120.5, NASA Space Flight Program and Project Management Requirements

c. NPR 8715.3, NASA General Safety Program Requirements

d. NPR 7123.1, NASA Systems Engineering Processes and Requirements

e. NPR 7150.2, NASA Software Engineering Requirements

f. NASA-STD-8719.13, Software Safety Standard

P.4 Applicable Documents

a. DCP-S-007, Software Assurance

b. 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

None

/S/	August 18, 2009
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 requirements for development software at Dryden. It includes the applicable software engineering requirements specified in NPR 7150.2. It also contains the requirements in NASA-STD-8719.13 that apply when software is classified as "safety critical". Finally, it includes requirements derived from RTCA DO-178, Software Considerations in Airborne Systems and Equipment Certification, to fill in those areas where Dryden's processes are more stringent.

1.2 DOCUMENT SCOPE

1.2.1 The requirements of this document cover software created or acquired by or for Dryden, 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.

1.2.2 The applicability of requirements in this standard to specific systems and subsystems is determined through the use of the software classifications described in Section 5, in conjunction with the Requirements Mapping Matrix.

1.2.3 This standard shall be applied to software development, maintenance, operations, management, acquisition, and assurance activities started after its effective date of issuance. Contracts that involve software development should include reference to this DPR.

1.2.4 This standard 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.

1.2.5 This standard does not supersede more stringent requirements imposed by individual NASA organizations and other Federal Government agencies.

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

1.3 DEFINITION OF SOFTWARE

1.3.1 For the purposes of this document, software is defined in NASA Standard 8739.8, Software Assurance Standard, Section 3.1, as computer programs, procedures, rules, and associated documentation and data pertaining to the development and operation of a computer system. Software includes programs and operational data contained in hardware (e.g., firmware, programmable logic, and programmable gate

arrays). This also includes COTS, GOTS, MOTS, reuse, legacy, and heritage software products and components.

1.3.2 Types of software include:

a. Application software: Word processors and other programs that perform productive tasks for users.

b. Firmware: Software loaded onto and resident within electronic components and/or peripherals.

c. Middleware: Software that controls and co-ordinates distributed systems.

d. System software: Software such as operating systems and drivers that interfaces with hardware to provide the necessary services for application software.

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 Note that NASA is currently working on a standard to address complex electronics, including Complex Programmable Logic Devices (CPLDs), Field Programmable Gate Arrays (FPGAs), and Application Specific Integrated Circuits (ASICs). Once that document is released, the center will reassess the need for a separate Dryden standard for these components.

1.4 SOFTWARE DOCUMENTATION AND THE PROJECT LIFECYCLE

1.4.1 DPR 7123.1-001, Chapter 1, provides a description of the project lifecycle as it applies to a typical Dryden project.

1.4.2 Table 1 below provides a mapping from this lifecycle to major project software activities and the artifacts associated with those activities. It also maps documentation requirements to Software Classification, which is captured in Chapter 2.

Phase	Event	Document(s)	S/W Classification				
			I	II	III	IV	S
A	PAG	Project Plan					x
B	SRR	Software Configuration Management Plan (Could be part of Project Configuration Management Plan)	x	x	x		x
		Software Assurance Plan/Software Quality Assurance Plan	x	x			x
		Software Development Plan	x	x	x		x
		Software Management Plan/Software Project Management Plan (Can be combined with Software Development Plan)	x	x	x		x
		Software Safety Plan					x
	PDR	Safety Analyses and Reports					x
		Top Level Software Design Documentation	x	x			x
		Software Requirements Specification	x	x	x		x
C	CDR	Detailed Software Design Documentation, including a Software Data Dictionary and Interface Design Description	x	x			x
		Operations and Maintenance Plan(s)	x	x			x
		Software Test Plan(s)	x	x	x		x
D	Integration & Test	Software Test Procedures	x	x			x
		Software Test Report(s)	x	x			x
E	ORR	User documentation and procedures, or Software Users Manual	x	x			x
All	Software Release	Software Version Description	x	x	x	x	x

Table 1 - Documentation Requirements and the Dryden Project Lifecycle

CHAPTER 2. RESPONSIBILITIES

The Designated Governing Authority (DGA) for the software engineering requirements described in this DPR begins with the Dryden Center Director per DPR 7123.1-001, Systems Engineering Requirements Document, Section 1.3. From there, authority is delegated to the Associate Director of Operations; Director for Management, Safety, and Mission Assurance Lead; or the Acquisition Management Officer based on the type of requirement. The general delegation strategy is listed below. The specific mapping of requirements to center DGA delegates is provided in Appendix D.

.2.1 DRYDEN CENTER DIRECTOR

2.1.1 The Dryden Center Director is the Designated Governing Authority for Center Level requirements dealing with Applicability and Scope, Best Practices, Expertise of Independent Technical Authority (ITA) Warrant Holders, Organizational Capability, Tailoring of Requirements, and Training. The Director Center Director is also the Designated Governing Authority for requirements covering legal compliance.
See Table D.1 in Appendix D for the specific list.

2.2 Associate Director For Mission Support

2.2.1 The Associate 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 Table D.2 in 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. See Tables D.2 and D.3 in Appendix D for the specific lists.

2.4. Director for Mission 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 Tables D.2 and D.4 in 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

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 the also the Designated Governing Authority for requirements covering the control of software loaded on aircraft. See Tables D.2 and D.5 in 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 Tables D.2 and D.6 in 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 Table D.7 in Appendix D for the specific list.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3. SOFTWARE CLASSIFICATION

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 one 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. (The use of Roman Numerals should reduce confusion with other software standards [NPR 7150.2, RTCA DO-178, etc.] that utilize alphabetic classifications.) Specifically, these categories are as follows.

a. CLASS I: CATASTROPHIC

b. CLASS II: CRITICAL

c. CLASS III: MINOR

d. CLASS IV: NEGLIGIBLE

3.1.2 Since the scope of this document goes beyond just flight software; these category definitions have been expanded to address other types of consequences, such as a software-related security breach or agency-wide loss of productivity. To incorporate the "Safety Critical" software requirements levied in NASA-STD-8719.13, an additional "S" will be added to the classification to denote safety critical software. For example, software that could cause a critical security breach would be classified as Class II. 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 PHA will review the software down to the Computer Software Configuration Item (CSCI) level, the level at which the 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.

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.

3.3 CRITERIA FOR SAFETY CRITICAL SOFTWARE

3.3.1 “Safety Critical” is defined in NASA-STD-8719.13, Section 3.1 as follows: “Any condition, event, operation, process, equipment, or system that possesses the potential of directly or indirectly causing harm to humans, damage to property external to the system, or damage to the environment”.

3.3.2 Software is considered safety-critical if it meets at least one of the following criteria per NASA-STD-8719.13, Section 4.1.1.2.

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

a) Causes or contributes to a safety-critical hazard.

b) Provides control or mitigation for a safety-critical hazard.

c) Controls safety-critical functions.

d) Processes safety-critical commands or data.

e) Detects and reports, or takes corrective action, if the system reaches a specific hazardous state.

f) Mitigates damage if a safety-critical hazard occurs.

g) Resides on the same system (processor) as safety-critical software.

2) Processes data or analyzes trends that lead directly to safety decisions (e.g., determining when to turn off power to a wind tunnel to prevent system destruction).

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

3.4 SOFTWARE CLASSIFICATIONS – SAFETY CRITICAL

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

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

1) Medical treatment for a minor injury or occupational illness (no lost time).

d. Class IV-S: Minimal

N/A

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 $1M

b. Class II: Critical

1) Long term project delay

2) Loss of some project critical data

3) Loss of confidentiality, integrity, and/or availability of information with an IT security category of “High” per NPR 2810.1, Section 7.2

4) Recovery/replacement cost equal to or greater than $250K, but less than $1M

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

4) Minor loss/damage to facility, system, equipment, or flight hardware

5) Recovery/replacement cost equal to or greater than $25K, but less than $250K.

6) Interruptions in the availability of Critical Data

7) Center-wide Productivity Impact

d. Class IV: Minimal

1) Productivity Impact to small number of users

3.6 CLASSIFICATION GUIDELINES

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

3.6.1 Destruction of Facility, Major System, or Vehicle

a. 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 $1,000,000 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:

1) Intentional destruction of a vehicle

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

3.6.2 Recovery/Replacement Costs

a. DCP-S-002 and NPR 8621.1 both provide criteria for recovery/replacement costs (for instance, $1,000,000 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.

3.6.3 Major Damage vs. Destruction

a. 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.

3.6.4 Data Criticality

a. IT security categories are defined in NPR 2810.1, Section 7.2. Projects must determine the criticality of their data with respect to their mission. Will there be other opportunities to collect that data? Are there other parameters that could be used to

compute, recreate, or approximate missing data? Is the data critical to meeting mission objectives?

3.6.5 Long Term Delay

a. 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.

3.6.6 Loss of Mission

a. 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 one and 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.

3.6.7 Interruptions in Availability

a. 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. For instance, failure of a control room display monitoring a critical airborne system would produce safety impacts that would ultimately dictate the criticality of that software system.

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 2).

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

c. These mitigations are verifiable and verified.

> 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.

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

Table 2: Architectural Considerations in Software Criticality Assessment

* Trained test pilots who possess the situational awareness to know when a hazard exists as well as the procedures, training, and time to mitigate that hazard can be considered elements of a system for flight test operations.

CHAPTER 4. SOFTWARE ENGINEERING REQUIREMENTS

This section includes the specific Dryden 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.

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

4.1 CENTER LEVEL SOFTWARE ENGINEERING REQUIREMENTS

NOTE: *All of the requirements listed in this section apply at the Center level. They are not applicable to individual projects or activities.*

4.1.1 Applicability and Scope

a. Effective Date **(R.0060)** – *See NPR 7150.2, SWE-001*

4.1.2 Organizational Capability and Improvement

a. Center Plan (R.0070) – See NPR 7150.2, SWE-003

b. SW Processes (R.0080) – See NPR 7150.2, SWE-005

4.1.3 Best Practices

a. Identify applicable practices **(R.0100)** – *See NPR 7150.2, SWE-099*

4.1.4 Training

a. Software engineering training **(R.0110)** – *See NPR 7150.2, SWE-100*

b. Software training plan **(R.0120)** – *See NPR 7150.2, SWE-101*

4.1.5 Software Plans

a. Center SW Training Plan **(R.0130)** – *See NPR 7150.2, SWE-107*

b. Center SW Engineering Improve Plan **(R.0140)** – *See NPR 7150.2, SWE-108*

4.1.6 Tailoring of Requirements

a. Alternate requirement request **(R.0150)** – *See NPR 7150.2, SWE-120*

b. Document approved alternate requirements **(R.0160)** – *See NPR 7150.2, SWE-121*

4.1.7 Expertise of ITA Warrant Authority(s)

a. Non-IT and Non-Business **(R.0170)** – *See NPR 7150.2, SWE-122*

b. IT infrastructure and Business **(R.0180)** – *See NPR 7150.2, SWE-123*

4.1.8 Compliance

a. Direction for Warrant Authority **(R.0190)** – *See NPR 7150.2, SWE-124*

b. Considerations for Waivers **(R.0200)** – *See NPR 7150.2, SWE-126*

c. Review of "P(Center)" **(R.0210)** – *See NPR 7150.2, SWE-127*

d. Compliance Records **(R.0220)** – *See NPR 7150.2, SWE-128*

4.1.9 Software Safety Requirements

4.1.9.1 Determination of Safety-Critical Software

a. SMA approves of the evaluation conclusions. **(R.0230)** – *See NASA-STD-8719.13, 4.1.1.5*

4.1.9.2 Certification Process

a. Establish a certification process for safety-critical software. Safety-critical software is certified prior to use or release. (R.0240) – See NASA-STD-8719.13, 5.14.1

b. Participate in program/project/facility certifications (R.0250) – See NASA-STD-8719.13, 5.14.2

c. 5.14.3-(a-g) are items to be evaluated for certification (R.0260) – See NASA-STD-8719.13, 5.14.3

d. All software hazards are identified. (R.0270) – See NASA-STD-8719.13, 5.14.3-a

e. All hazard controls that require software implementation are identified. (R.0280) – See NASA-STD-8719.13, 5.14.3-b

f. All software safety requirements and elements are identified and tracked. (R.0290) – See NASA-STD-8719.13, 5.14.3-c

g. All software safety requirements and elements have been successfully validated, or waivers/deviations have been approved. (R.0300) – See NASA-STD-8719.13, 5.14.3-d

h. All software safety requirements and elements have been properly verified, or waivers/deviations have been approved. (R.0310) – See NASA-STD-8719.13, 5.14.3-e

i. All discrepancies in safety-critical software have been dispositioned with the safety organization's concurrence, per the certification process. (R.0320) – See NASA-STD-8719.13, 5.14.3-f

j. All operational workarounds associated with discrepancies in safety-critical software have the concurrence of the Center or Program safety organization, per the certification process. (R.0330) – See NASA-STD-8719.13, 5.14.3-g

k. Approve the results and reports prior to acceptance of the software and the system, with review and certification provided by SMA. (R.0340) – See NASA-STD-8719.13, 5.14.5

4.1.9.3 Operational Use of Software

a. This Standard applies to safety-critical software that has been released for operations. (R.0350) – See NASA-STD-8719.13, 7.1

b. When a system or facility is retired, this Standard no longer applies. (R.0360) – See NASA-STD-8719.13, 7.4

c. A retirement plan will address the safe termination of operations, decommissioning, and retirement of the system or facility. (R.0370) – See NASA-STD-8719.13, 7.4.1

4.1.9.4 Waivers/ Deviations

a. Keep copies of all variances to safety requirements. (R.0380) – See NASA-STD-8719.13, 5.15.3

4.2 PROJECT LEVEL SOFTWARE ENGINEERING REQUIREMENTS

4.2.1 Compliance with Laws, Policies, and Requirements

a. SW Safety **(R.0090)** – See NPR 7150.2, SWE-023

b. SW Disclosures **(R.0390)** – See NPR 7150.2, SWE-007

c. Export Control **(R.0400)** – See NPR 7150.2, SWE-008

d. External Release **(R.0410)** – See NPR 7150.2, SWE-009

e. Security **(R.0420)** – See NPR 7150.2, SWE-010

f. Disabilities - Reasonable Accommodation **(R.0430)** – See NPR 7150.2, SWE-011

g. Disabilities - Section 508 Compliance **(R.0440)** – See NPR 7150.2, SWE-012

4.2.2 Software Life Cycle Planning

a. SW Plan **(R.0450)** – See NPR 7150.2, SWE-013

b. Execute Plan **(R.0460)** – See NPR 7150.2, SWE-014

c. Cost Estimation **(R.0470)** – See NPR 7150.2, SWE-015

d. Schedule **(R.0480)** – See NPR 7150.2, SWE-016

e. Training **(R.0490)** – See NPR 7150.2, SWE-017

f. Reviews **(R.0500)** – See NPR 7150.2, SWE-018

g. Life Cycle **(R.0510)** – See NPR 7150.2, SWE-019

h. SW Classification **(R.0520)** – See NPR 7150.2, SWE-020

i. SW Classification changes **(R.0530)** – See NPR 7150.2, SWE-021

j. SW Assurance **(R.0540)** – See NPR 7150.2, SWE-022

k. Plan Tracking **(R.0550)** – See NPR 7150.2, SWE-024

l. Corrective Action **(R.0560)** – See NPR 7150.2, SWE-025

m. Changes **(R.0570)** – See NPR 7150.2, SWE-026

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

a. COTS, GOTS, MOTS **(R.0580)** – *See NPR 7150.2, SWE-027*

4.2.4 Project Formulation Requirements

a. Supplier Selection **(R.0600)** – *See NPR 7150.2, SWE-035*

b. Acquisition planning **(R.0610)** – *See NPR 7150.2, SWE-038*

4.2.5 Software Safety Requirements

4.2.5.1 Program, Project, Facility Management

a. Consult with software safety personnel when acquiring safety-critical software (R.0620) – See NASA-STD-8719.13, 5.1.2.1.1

b. Periodically evaluate the system for safety-critical software (R.0630) – See NASA-STD-8719.13, 5.1.2.1.2

c. Provides adequate resources to the software safety program. (R.0640) – See NASA-STD-8719.13, 5.1.2.1.3

d. Assign personnel for the software safety program (R.0650) – See NASA-STD-8719.13, 5.1.2.1.4

e. Work with SMA management to resolve conflicts (R.0660) – See NASA-STD-8719.13, 5.1.2.1.5

f. Plan and execute software safety throughout the entire software life cycle. (R.0670) – See NASA-STD-8719.13, 5.1.2.2

g. Integrate software safety with system safety and software development. (R.0680) – See NASA-STD-8719.13, 5.1.2.3

h. Create a process to document, trace, communicate, and close software safety concerns (R.0690) – See NASA-STD-8719.13, 5.1.2.4

4.2.5.2 Software Safety Personnel

a. Assign S/W safety manager to develop and implement S/W safety processes and plans (R.0700) – See NASA-STD-8719.13, 5.1.3.1

b. Assign personnel to perform software safety activities (R.0710) – See NASA-STD-8719.13, 5.1.3.2

4.2.5.3 Other Personnel

a. Assign software assurance to verify that software safety is planned, approved, and implemented. (R.0720) – See NASA-STD-8719.13, 5.1.4.1

4.2.5.4 Documentation Requirements

a. Define the change and approval process for software safety related portions of all project documents. (R.0730) – See NASA-STD-8719.13, 5.6.2

4.3 SYSTEM LEVEL SOFTWARE ENGINEERING REQUIREMENTS

4.3.1 Software Verification and Validation

a. Verification planning **(R.0740)** – *See NPR 7150.2, SWE-028*

b. Validation planning **(R.0750)** – *See NPR 7150.2, SWE-029*

c. Verification results **(R.0760)** – *See NPR 7150.2, SWE-030*

d. Validation results **(R.0770)** – *See NPR 7150.2, SWE-031*

4.3.2 Project Formulation Requirements

a. Options for Acquisition **(R.0780)** – *See NPR 7150.2, SWE-033*

b. Acceptance Criteria **(R.0790)** – *See NPR 7150.2, SWE-034*

c. SW processes and tasks **(R.0800)** – *See NPR 7150.2, SWE-036*

d. Milestones **(R.0810)** – *See NPR 7150.2, SWE-037*

4.3.3 Software Requirements

a. Documented Requirements **(R.0820)** – See NPR 7150.2, SWE-049

b. SW requirements **(R.0830)** – See NPR 7150.2, SWE-050

c. Flow-down and derived req. **(R.0840)** – See NPR 7150.2, SWE-051

d. Bi-directional trace **(R.0850)** – See NPR 7150.2, SWE-05
e. 2
f. Manage req. change **(R.0860)** – See NPR 7150.2, SWE-053

g. Corrective action **(R.0870)** – See NPR 7150.2, SWE-054

h. Requirements Validation **(R.0880)** – See NPR 7150.2, SWE-055

i. 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. **(R.0882)** – (DO-178, 6.3.1g)

j. Software 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. **(R.0924)** – (DO-178, 6.3.3a)

k. Review of Software Architecture data and control flow: The project shall review the proposed software architecture to ensure the correct data flow and control flow

relationships exist between the components of that architecture. **(R.0926)** – (DO-178, 6.3.3b)

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

4.3.4 Software Design

a. Bi-directional 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. Maintain Traceability **(R.0930)** – See NPR 7150.2, SWE-064

b. Coding standards **(R.0940)** – See NPR 7150.2, SWE-061

c. Unit test **(R.0950)** – See NPR 7150.2, SWE-062

d. Version Description **(R.0960)** – See NPR 7150.2, SWE-063

4.3.6 Software Testing

a. Models, simulations, tools **(R.0970)** – *See NPR 7150.2, SWE-070*

b. Plan, procedures, reports **(R.0980)** – *See NPR 7150.2, SWE-065*

c. Perform testing **(R.0990)** – *See NPR 7150.2, SWE-066*

d. Test for compliance **(R.1000)** – *See NPR 7150.2, SWE-067*

e. Performance of Structural Coverage Analysis: The project shall perform structural coverage analysis after requirement verification is performed to identify code that was not exercised during that testing. **(R.1002)** – *(DO-178, 6.4.4.2)*

f. Utilize Structural Coverage Analysis to identify additional testing requirements: The project shall utilize the results of the structural coverage analysis to identify and perform additional software testing to ensure that a) every decision in the program has taken all possible outcomes at least once, b) every condition in a decision in the program has taken all possible outcomes at least once, and c) every condition in a decision has been shown to independently affect that decision's outcome. **(R.1004)** – *(DO-178, 6.4.4.3)*

g. Dead Code Identification and Removal: The project shall identify and remove any dead code identified during the structural coverage analysis and assess the effect and the need for reverification based on that change. **(R.1006)** – *(DO-178, 6.4.4.3c)*

h. Deactivated Code Analysis: The project shall identify any deactivated code (code which 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. **(R.1008)** – *(DO-178, 6.4.4.3d)*

i. Evaluate test results **(R.1010)** – *See NPR 7150.2, SWE-068*

j. Document defect and track **(R.1020)** – *See NPR 7150.2, SWE-069*

k. Update plans and procedures **(R.1030)** – *See NPR 7150.2, SWE-071*

l. Maintain Traceability **(R.1040)** – *See NPR 7150.2, SWE-072*

m. Platform or Hi-Fidelity simulation. **(R.1050)** – *See NPR 7150.2, SWE-073*

4.3.7 Software Operations, Maintenance, and Retirement

a. Document maintenance. plans **(R.1060)** – *See NPR 7150.2, SWE-074*

b. Plan operations, maintenance, and retirement (R.1070) – See NPR 7150.2, SWE-075

c. Implement plans (R.1080) – See NPR 7150.2, SWE-076

d. Deliver software product (R.1090) – See NPR 7150.2, SWE-077

e. As-built documentation **(R.1100)** – *See NPR 7150.2, SWE-078*

4.3.8 Peer Reviews/Inspections

a. Requirements and Test Plans **(R.1110)** – *See NPR 7150.2, SWE-087*

b. Checklist, criteria, and tracking **(R.1120)** – *See NPR 7150.2, SWE-088*

c. SW Inspection/Peer Review **(R.2740)** – *See NPR 7150.2, SWE-119*

d. Basic measurements **(R.3140)** – *See NPR 7150.2, SWE-089*

4.3.9 Software Safety Requirements

4.3.9.1 Determination of Safety-Critical Software

a. If the system is safety-critical, evaluate the software. (R.1130) – See NASA-STD-8719.13, 4.1.1

b. Assume all software is safety-critical until proven otherwise. (R.1140) – See NASA-STD-8719.13, 4.1.1.1

c. Use the criteria in this section to determine if the software is safety-critical (R.1150) – See NASA-STD-8719.13, 4.1.1.2

d. Evaluate software during project planning (R.1160) – See NASA-STD-8719.13, 4.1.1.3

e. Document the results of the evaluation (R.1170) – See NASA-STD-8719.13, 4.1.1.4

f. Evaluate all software in the system (R.1180) – See NASA-STD-8719.1B, 4.1.2

4.3.9.2 Program, Project, Facility Management

a. Include software safety in project planning (R.1190) – See NASA-STD-8719.13, 5.1.2.1

4.3.9.3 Software Safety Planning

a. Plan and conduct safety program reviews (R.1200) – See NASA-STD-8719.13, 5.2.1.1

4.3.9.4 Off-the-shelf Software

a. Evaluate all off-the-shelf and reused software for its impact safety-critical functions. (R.1210) – See NASA-STD-8719.13, 5.12.1

b. Analyze safety-critical OTS and reused software for its ability to meet required safety functions, any safety impact of extra functionality, and interfaces to developed code. (R.1220) – See NASA-STD-8719.13, 5.12.1.1

c. Analyze the interactions of COTS software components with the developed software and any other COTS software in the system. (R.1230) – See NASA-STD-8719.13, 5.12.1.2

d. Verify safety-critical OTS or reused software to the same level required of in-house developed software to the extent possible. (R.1240) – See NASA-STD-8719.13, 5.12.1.3

4.3.9.5 Waivers/ Deviations

a. Request a waiver/deviation if a requirement cannot be met. (R.1250) – See NASA-STD-8719.13, 5.15.1

b. Document in a written request the justification to support the waiver/deviation. **(R.1260)** – *See NASA-STD-8719.13, 5.15.2*

4.3.9.6 Software Safety Requirements and Analysis

a. Create software safety requirements and include them in the software requirements specification. (R.1270) – See NASA-STD-8719.13, 6.1.1

b. Derive software safety requirements from system safety, environmental, interface, vehicle, and facility requirements; standards, program specification, system hazard reports and analyses. (R.1280) – See NASA-STD-8719.13, 6.1.1.1

c. Clearly identify software safety requirements in the software requirements specification. (R.1290) – See NASA-STD-8719.13, 6.1.1.2

d. Express and structure software safety requirements that are clear, precise, unequivocal, verifiable, testable, maintainable, and feasible. (R.1300) – See NASA-STD-8719.13, 6.1.1.3

e. Include the modes or states of operation under which software safety requirements are valid or not applicable. (R.1310) – See NASA-STD-8719.13, 6.1.1.4

f. Include hardware and software safety-related constraints in software requirements document. **(R.1320)** – *See NASA-STD-8719.13, 6.1.1.5*

4.3.9.7 Software Design and Safety Analysis

a. Incorporate all functional software safety requirements into the software design. (R.1330) – See NASA-STD-8719.13, 6.2.1

b. Identify safety design features and methods in the software design. (R.1340) – See NASA-STD-8719.13, 6.2.1.1

c. Allow for software safety features and requirements be thoroughly tested in the software design. (R.1350) – See NASA-STD-8719.13, 6.2.1.2

d. Designate design elements that implement safety-critical requirements as safety-critical. (R.1360) – See NASA-STD-8719.13, 6.2.1.3

e. Clearly identify all safety-critical design elements in software design documentation. (R.1370) – See NASA-STD-8719.13, 6.2.1.3.1

f. Modularize the safety-related aspects of the design in the software design. (R.1380) – See NASA-STD-8719.13, 6.2.1.4

4.3.9.8 Software Implementation and Safety Analysis

a. Verify unit testing and data verification is completed before the unit is integrated. (R.1390) – See NASA-STD-8719.13, 6.3.3

4.3.9.9 Software Test and Safety Analysis

a. Verify all functional software safety requirements and safety-critical software elements by testing. (R.1400) – See NASA-STD-8719.13, 6.4.1

b. Verify via test that system hazards related to software have been eliminated or controlled to an acceptable level of risk. (R.1410) – See NASA-STD-8719.13, 6.4.1.1

c. Include software safety testing in unit level tests and component level tests. (R.1420) – See NASA-STD-8719.13, 6.4.1.2

d. Configuration control unit test simulators, test drivers, stubs, and test data. (R.1430) – See NASA-STD-8719.13, 6.4.1.2.1

e. Configuration control component test simulators, test drivers, stubs, and test data. (R.1440) – See NASA-STD-8719.13, 6.4.1.2.2

f. Document the results of unit level and component level tests, plus the test procedures, simulators, test suites, drivers, stubs, and data used. (R.1450) – See NASA-STD-8719.13, 6.4.1.2.3

g. Include software safety testing within system and acceptance tests. (R.1460) – See NASA-STD-8719.13, 6.4.1.3

h. Verify the correct and safe operation of the software in conjunction with system hardware and operator inputs prior to system acceptance. (R.1470) – See NASA-STD-8719.13, 6.4.1.3.1

i. Verify the correct and safe operation of the system in the presence of failures and faults. (R.1480) – See NASA-STD-8719.13, 6.4.1.3.2

j. Use safety analyses to determine which failures to test for, how many, and in what combinations. (R.1490) – See NASA-STD-8719.13, 6.4.1.3.3

k. Verify the correct and safe operation of the system under system load, stress, and off-nominal conditions. (R.1500) – See NASA-STD-8719.13, 6.4.1.3.4

l. Verify the system operates correctly and safely in all anticipated operational and off-nominal configurations. (R.1510) – See NASA-STD-8719.13, 6.4.1.3.5

m. Analyze newly identified hazardous states or contributors prior to software delivery or use. (R.1520) – See NASA-STD-8719.13, 6.4.1.4

n. Evaluate, inspect, or demonstrate requirements if testing is not feasible. (R.1530) – See NASA-STD-8719.13, 6.4.2

o. Record the rationale for choosing evaluation, inspection, or demonstration over test. (R.1540) – See NASA-STD-8719.13, 6.4.2.1

p. Record the evaluation, inspection, or demonstration methodology. (R.1550) – See NASA-STD-8719.13, 6.4.2.2

q. Get the software safety engineer's concurrence with both the rationale and the methodology. (R.1560) – See NASA-STD-8719.13, 6.4.2.3

4.3.9.10 Operational Use of Software

a. When operational software is changed, specify, develop, analyze, and test all safety-critical software. (R.1570) – See NASA-STD-8719.13, 7.2

4.4 DEVELOPER LEVEL SOFTWARE ENGINEERING REQUIREMENTS

4.4.1 Software Implementation

a. Design --> code **(R.1580)** – *See NPR 7150.2, SWE-060*

4.4.2 Software Safety Requirements

4.4.2.1 Software Implementation and Safety Analysis

a. Implement all software safety design features and methods in the software code. (R.1590) – See NASA-STD-8719.13, 6.3.1

b. Incorporate software coding standards that prohibit unsafe language features and require commenting of safety-critical source code. (R.1600) – See NASA-STD-8719.13, 6.3.1.1

c. Utilize software coding standard in developing code. (R.1610) – See NASA-STD-8719.13, 6.3.1.2

4.5 SYSTEM SAFETY SOFTWARE ENGINEERING REQUIREMENTS

4.5.1 Risk Management

a. Continuous risk management **(R.1620)** – *See NPR 7150.2, SWE-086*

4.5.2 Software Safety Requirements

4.5.2.1 Software and System Safety

a. Participate in system safety analyses (R.1630) – See NASA-STD-8719.13, 4.2.1

b. Evaluate hazards for software's contribution (cause, control, etc.). (R.1640) – See NASA-STD-8719.13, 4.2.1.1

c. Conduct software safety analyses; coordinate with the system safety analyses. (R.1650) – See NASA-STD-8719.13, 4.2.1.2

d. Create software safety requirements. (R.1660) – See NASA-STD-8719.13, 4.2.2

e. Document software safety requirements, hazard contributors, and controls. Software safety plan points to this document. (R.1670) – See NASA-STD-8719.13, 4.2.2.1

4.5.2.2 Software Safety Personnel

a. Communicate software safety concerns directly to the project manager. (R.1680) – See NASA-STD-8719.13, 5.1.3.1.1

b. Elevate software safety concerns that cannot be resolved within the project. (R.1690) – See NASA-STD-8719.13, 5.1.3.1.2

c. Assure software safety risks are captured, addressed, and managed as part of risk management processes. (R.1700) – See NASA-STD-8719.13, 5.1.3.1.3

d. Participate in change control board for software modifications. (R.1710) – See NASA-STD-8719.13, 5.1.3.1.4

e. Brief management on the selection of off-the-shelf or previously created (reused) software used in this system. (R.1720) – See NASA-STD-8719.13, 5.1.3.1.5

f. Provide inputs to management regarding contractor requirements for safety-critical software. (R.1730) – See NASA-STD-8719.13, 5.1.3.1.6

g. Analyze and report software safety nonconformances to appropriate personnel. (R.1740) – See NASA-STD-8719.13, 5.1.3.2.1

h. Review system hazard analyses for changes that impact the software subsystem. (R.1750) – See NASA-STD-8719.13, 5.1.3.2.2

i. Inform system safety personnel of changes in safety-critical software. (R.1760) – See NASA-STD-8719.13, 5.1.3.2.3

j. Support the system safety review process. (R.1770) – See NASA-STD-8719.13, 5.1.3.2.4

k. Participate in project reviews. (R.1780) – See NASA-STD-8719.13, 5.1.3.2.5

4.5.2.3 Software Safety Planning

a. Perform software safety assessment and planning for each software acquisition, maintenance activity, or change to legacy systems. (R.1790) – See NASA-STD-8719.13, 5.2.1

b. Start software safety planning early enough to affect the software development and assurance activities. (R.1800) – See NASA-STD-8719.13, 5.2.2

c. Create a Software Safety Plan. (R.1810) – See NASA-STD-8719.13, 5.2.3

d. Review the Software Safety Plan periodically. (R.1820) – See NASA-STD-8719.13, 5.2.7

4.5.2.4 Software Life Cycles

a. Perform software safety activities throughout the software development life cycle. (R.1830) – See NASA-STD-8719.13, 5.5.2

b. Continue software safety activities in the operational phase. (R.1840) – See NASA-STD-8719.13, 5.5.3

4.5.2.5 Traceability

a. Create a tracing system that maps software safety requirements to system hazards and traces the flow down of to design, implementation, and test. (R.1850) – See NASA-STD-8719.13, 5.7.1

b. Coordinate the software tracing system with the system-level hazard tracking system. (R.1860) – See NASA-STD-8719.13, 5.7.1.1

c. Put the tracing system under configuration control. (R.1870) – See NASA-STD-8719.13, 5.7.2

d. Review the tracing system reports and outputs. (R.1880) – See NASA-STD-8719.13, 5.7.3

4.5.2.6 Software Assurance Activities

a. Complete all assurance activities prior to acceptance or closure of any software-related system-level hazards. (R.1890) – See NASA-STD-8719.13, 5.10.1

b. Coordinate software safety tasks with software assurance. (R.1900) – See NASA-STD-8719.13, 5.10.2

4.5.2.7 Tool Support and Approval

a. Ensure sufficient safety testing and analysis is performed when a project tool changes (R.1910) – See NASA-STD-8719.13, 5.11.1.2.2

b. Approve the safety analysis and testing approach for tool verification. (R.1920) – See NASA-STD-8719.13, 5.11.2

4.5.2.8 Certification Process

a. Present the software safety process and results to an appropriate safety panel for certification. (R.1930) – See NASA-STD-8719.13, 5.14.4

4.5.2.9 Software Safety Requirements and Analysis

b. Analyze software safety requirements. (R.1940) – See NASA-STD-8719.13, 6.1.2

c. Ensure the analysis method or procedure meets the following requirements (a-e) and is documented. (R.1950) – See NASA-STD-8719.13, 6.1.2.1

d. Verify all software safety requirements meet the requirements of section 6.1.1 and subsections. (R.1960) – See NASA-STD-8719.13, 6.1.2.1-a

e. Examine the software safety requirements for ambiguities, inconsistencies, omissions, and undefined conditions. (R.1970) – See NASA-STD-8719.13, 6.1.2.1-b

f. Verify all software safety requirements are traceable to higher-level requirements or external standards. (R.1980) – See NASA-STD-8719.13, 6.1.2.1-c

g. Verify software safety requirements provide adequate response to potential failures. (R.1990) – See NASA-STD-8719.13, 6.1.2.1-d

h. Verify software safety requirements include positive measures to prevent potential problems and implement required “must work” functions. (R.2000) – See NASA-STD-8719.13, 6.1.2.1-e

i. Document results of the analysis. Give any newly identified hazards to system safety. (R.2010) – See NASA-STD-8719.13, 6.1.2.2

j. Document project-level resolution for improperly decomposed requirements. (R.2020) – See NASA-STD-8719.13, 6.1.2.3

k. Include results of the software safety requirements analysis at project formal reviews and system-level safety reviews. (R.2030) – See NASA-STD-*8719.13, 6.1.2.4*

4.5.2.10 Software Design and Safety Analysis

a. Analyze the software design. (R.2040) – See NASA-STD-8719.13, 6.2.2

b. Document the analysis methodology. (R.2050) – See NASA-STD-8719.13, 6.2.2.1

c. Ensure documented analysis method or procedure meets the following requirements (a-f). (R.2060) – See NASA-STD-8719.13, 6.2.2.2

d. Verify software design meets the requirements of section 6.2.1 and subsections. (R.2070) – See NASA-STD-8719.13, 6.2.2.2-a

e. Verify design does not compromise any safety controls or processes. Ensure that any additional hazard cause or contribution is documented. The design maintains the system in a safe state during all modes of operation. (R.2080) – See NASA-STD-8719.13, 6.2.2.2-b

f. Verify safety features are adequate for their function. (R.2090) – See NASA-STD-8719.13, 6.2.2.2-c

g. Determine design features necessary to prevent, mitigate, or control failures and faults, and the partitioning of safety features between hardware and software. (R.2100) – See NASA-STD-8719.13, 6.2.2.2-d

h. Verify that any partitioning or isolation methods adequately isolate the safety-critical design elements from those that are non-safety-critical. (R.2110) – See NASA-STD-8719.13, 6.2.2.2-e

i. Verify all safety-critical design elements are traceable to software safety requirements, and vice versa. (R.2120) – See NASA-STD-8719.13, 6.2.2.2-f

j. Provide the documented results of the analysis and any newly identified hazards to system safety. (R.2130) – See NASA-STD-8719.13, 6.2.2.3

k. Include the results of the software safety design analysis at project formal reviews and system-level safety reviews. (R.2140) – See NASA-STD-8719.13, 6.2.2.4

4.5.2.11 Software Implementation and Safety Analysis

a. Analyze the software implementation (e.g., code). (R.2150) – See NASA-STD-8719.13, 6.3.2

b. Document the analysis methodology. (R.2160) – See NASA-STD-8719.13, 6.3.2.1

c. Ensure the analysis method or procedure meets the following requirements (a-e) and is documented. (R.2170) – See NASA-STD-8719.13, 6.3.2.2

d. Verify safety-critical software code and data meets the requirements of section 6.3.1 and subsections. (R.2180) – See NASA-STD-8719.13, 6.3.2.2-a

e. Verify design safety features and methods are correctly implemented in the software code. (R.2190) – See NASA-STD-8719.13, 6.3.2.2-b

f. Verify the software maintains the system in a safe state during all modes of operation and does not compromise any safety controls or processes, nor create any additional hazards. (R.2200) – See NASA-STD-8719.13, 6.3.2.2-c

g. Ensure code and data verification activities include software safety requirements, if a requirement can be verified at this level. (R.2210) – See NASA-STD-8719.13, 6.3.2.2-d

h. Verify all safety-critical code units are traceable to safety-critical design elements. (R.2220) – See NASA-STD-8719.13, 6.3.2.2-e

i. Provide the documented results of the analysis and any newly identified hazards to system safety. (R.2230) – See NASA-STD-8719.13, 6.3.2.3

j. Include the results of the software safety design analysis at project formal reviews and system-level safety reviews. (R.2240) – See NASA-STD-8719.13, 6.3.2.4

4.5.2.12 Software Test and Safety Analysis

a. Analyze results from the software and system test process or requirements verification process. (R.2250) – See NASA-STD-8719.13, 6.4.3

b. Document the analysis methodology. (R.2260) – See NASA-STD-8719.13, 6.4.3.1

c. Ensure the analysis method or procedure meets the following requirements (a-d) and is documented. (R.2270) – See NASA-STD-8719.13, 6.4.3.2

d. Verify software and system tests data meet the requirements of section 6.4.1 and subsections. (R.2280) – See NASA-STD-8719.13, 6.4.3.2-a

e. Verify that the requirements verification evaluation, inspection, or demonstration data meet the requirements of section 6.4.2 and subsections. (R.2290) – See NASA-STD-8719.13, 6.4.3.2-b

f. Verify test coverage analysis shows that all safety requirements, functions, controls, and processes have been completely covered. (R.2300) – See NASA-STD-8719.13, 6.4.3.2-c

g. Verify software safety requirements have been tested, or evaluated, inspected, or demonstrated. (R.2310) – See NASA-STD-8719.13, 6.4.3.2-d

h. Verify all software safety functions are correctly performed and the software system does not perform unintended functions. (R.2320) – See NASA-STD-8719.13, 6.4.3.2-e

i. Document the results of the analysis and provide any newly identified hazards to system safety. (R.2330) – See NASA-STD-8719.13, 6.4.3.3

j. Document and report improperly implemented requirements for project-level resolution. (R.2340) – See NASA-STD-8719.13, 6.4.3.4

4.5.2.13 Operational Use of Software

a. Present the results of the software safety design analysis at project formal reviews and system-level safety reviews. (R.2350) – See NASA-STD-8719.13, 6.4.3.5

b. Evaluate proposed changes for their impact on system safety. (R.2360) – See NASA-STD-8719.13, 7.2.1

c. Assess the amount of regression testing needed. (R.2370) – See NASA-STD-8719.13, 7.2.1.1

d. Concur on any changes to basic, as built, or approved upgrades of the operational software. (R.2380) – See NASA-STD-8719.13, 7.2.1.2

e. Review updates to user manuals and procedures for safety impacts, and verify software safety related operational workarounds are properly documented. (R.2390) – See NASA-STD-8719.13, 7.3.2

4.6 CONFIGURATION MANAGEMENT REQUIREMENTS

4.6.1 Software Configuration Management

a. Develop CM plan. **(R.2400)** – *See NPR 7150.2, SWE-079*

b. Track and evaluate changes. **(R.2410)** – *See NPR 7150.2, SWE-080*

c. Identify S/W configuration items. **(R.2420)** – *See NPR 7150.2, SWE-081*

d. Authorize changes. **(R.2430)** – *See NPR 7150.2, SWE-082*

e. Maintain records. **(R.2440)** – *See NPR 7150.2, SWE-083*

f. Perform configuration audits. **(R.2450)** – *See NPR 7150.2, SWE-084*

g. Implement procedures. **(R.2460)** – *See NPR 7150.2, SWE-085*

4.6.1.1 Loading Flight Software

a. Labeling of Software Media: All software media (tape, disk, or chip) shall be identified and physically labeled at the time of production. (R.2461) – (DCP-S-007, p27)

b. Version Description Document: Prior to installation on the aircraft, a Version Description Document (VDD) shall be produced. (R.2462) – (DCP-S-007 p27)

c. Flight Media Release Form included in Version Description Document: The Version Description Document shall contain a Flight Media Release 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. (R.2463) – (DCP-S-007 p27)

d. Flight Software Installation Procedure: A procedure shall be written for flight software installation into the aircraft computer and for verification of correct loading. (R.2464) – (DCP-S-007 p28)

e. 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 Flight Media Release document. (R.2465) – (DCP-S-007 p28)

f. 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. (R.2466) – (DCP-S-007 p28)

4.6.2 Software Safety Requirements

4.6.2.1 Other Personnel

a. Perform configuration change control, status accounting, and change verification of safety-critical software requirements and software elements. (R.2470) – See NASA-STD-8719.13, 5.1.4.2

4.6.2.2 Discrepancy and Problem Reporting and Tracking

a. Create closed-loop tracking of discrepancies, problems, and failures. (R.2480) – See NASA-STD-8719.13, 5.8.1

b. Trace identified safety-critical software problems back to the system-level hazard involved. (R.2490) – See NASA-STD-8719.13, 5.8.1.1

c. Approve safety-critical discrepancy report closures. (R.2500) – See NASA-STD-8719.13, 5.8.1.2

d. Regularly review all discrepancy reports for safety impacts. (R.2510) – See NASA-STD-8719.13, 5.8.2

e. Evaluate software changes for potential safety impact. (R.2520) – See NASA-STD-8719.13, 5.8.3

4.6.2.3 Software Configuration Management Activities

a. Manage configuration of software, documentation, and associated data. (R.2530) – See NASA-STD-8719.13, 5.9.1

b. Maintain all baselined safety-critical software and associated documentation, simulators, models, test suites, data, etc. (R.2540) – See NASA-STD-8719.13, 5.9.1.1

c. Provide and document the release of safety-critical software. (R.2550) – See NASA-STD-8719.13, 5.9.1.2

d. Evaluate all changes, modifications, and patches made to safety-critical requirements, design, code, systems, equipment, test plans, procedures, simulators, models, test suites, or criteria. (R.2560) – See NASA-STD-8719.13, 5.9.2

e. Approve changes to baselined safety-critical software. (R.2570) – See NASA-STD-8719.13, 5.9.2.1

f. Track and control incremental changes to the safety-critical software and its release to operations. (R.2580) – See NASA-STD-8719.13, 5.9.3

g. Maintain configuration control of routine reconfigurations and changes to operational software (R.2590) – See NASA-STD-8719.13, 5.9.3.1

4.7 DOCUMENTATION REQUIREMENTS

4.7.1 Software Plans

4.7.1.1 Configuration Management Plan

a. SW Configuration Mgt. Plan (R.2610) – See NPR 7150.2, SWE-103

4.7.1.2 Software Assurance Plan

a. SW Assurance Plan (R.2640) – See NPR 7150.2, SWE-106

4.7.1.3 Software Development Plan

a. SW Development/Mgt. Plan (R.2600) – See NPR 7150.2, SWE-102

4.7.1.4 Software Maintenance Plan

a. SW Maintenance Plan (R.2630) – See NPR 7150.2, SWE-105

4.7.1.5 Test Plans

a. SW Test Plan (R.2620) – See NPR 7150.2, SWE-104

4.7.2 Software Design Documents

a. SW Requirements Spec. **(R.2650)** – *See NPR 7150.2, SWE-109*

b. SW Data Dictionary **(R.2660)** – *See NPR 7150.2, SWE-110*

c. SW Design Description **(R.2670)** – *See NPR 7150.2, SWE-111*

d. Interface Design Description **(R.2680)** – *See NPR 7150.2, SWE-112*

e. SW Change Request/ Problem Report **(R.2690)** – *See NPR 7150.2, SWE-113*

f. SW Test Procedures **(R.2700)** – *See NPR 7150.2, SWE-114*

g. SW Users Manual **(R.2710)** – *See NPR 7150.2, SWE-115*

h. SW Version Description **(R.2720)** – *See NPR 7150.2, SWE-116*

4.7.3 Software Reports

a. SW Test Report **(R.2730)** – *See NPR 7150.2, SWE-118*

4.7.4 Compliance

a. Compliance Matrix **(R.2750)** – *See NPR 7150.2, SWE-125*

4.7.5 Software Safety Requirements

4.7.5.1 Software Development Plan

a. Document the resource requirements and allocation for software safety tasks. (R.2760) – See NASA-STD-8719.13, 5.4.1

b. Define approach to preventing software tools from introducing hazards. (R.2770) – See NASA-STD-8719.13, 5.11.1

c. Identify and assess project tools that could potentially impact safety-critical software and define mitigation strategies if necessary. (R.2780) – See NASA-STD-8719.13, 5.11.1.1

d. Document how project tools are selected, approved, and controlled. (R.2790) – See NASA-STD-8719.13, 5.11.1.2

e. Document how approved tools are upgraded, what happens if an approved tool is no longer approved, and what limitations are imposed on tool use. (R.2800) – See NASA-STD-8719.13, 5.11.1.2.1

4.7.5.2 Software Safety Plan

a. Cross-reference safety activities that are in multiple plans. (R.2810) – See NASA-STD-8719.13, 5.2.3.1

b. Put the Software Safety Plan under configuration control. (R.2820) – See NASA-STD-8719.13, 5.2.3.2

c. Describe how the requirements of this Standard will be implemented. (R.2830) – See NASA-STD-8719.13, 5.2.4

d. Include activities, schedule, personnel, methods, and resulting products. (R.2840) – See NASA-STD-8719.13, 5.2.5

e. Define how system safety, software assurance, software development, and the Center or Program SMA organization works together. (R.2850) – See NASA-STD-8719.13, 5.2.6

f. Describe the role of IV&V and how IV&V will work with the software safety program and personnel. (R.2860) – See NASA-STD-8719.13, 5.2.6.1

g. Describe how safety-critical requirements are generated, implemented, tracked, and verified. (R.2870) – See NASA-STD-8719.13, 5.2.6.2

h. Define the procedures for resolving software safety concerns and recommendations. (R.2880) – See NASA-STD-8719.13, 5.2.6.3

i. Describe how software safety and project schedules are synchronized. (R.2890) – See NASA-STD-8719.13, 5.2.6.4

j. Specify the number and schedule of software safety assurance audits. (R.2900) – See NASA-STD-8719.13, 5.2.6.5

k. Document the conditions requiring software safety engineers to review a situation and proposed solutions or upgrades. (R.2910) – See NASA-STD-8719.13, 5.2.6.6

l. Define who monitors system during operation and what procedures are followed when they feel safety may be threatened. (R.2920) – See NASA-STD-8719.13, 5.2.6.7

m. Describe the training requirements for all project software safety roles. (R.2930) – See NASA-STD-8719.13, 5.3.1

n. Describe how software safety is integrated with the chosen software life cycle. (R.2940) – See NASA-STD-8719.13, 5.5.1

o. List the documents (and associated content) that are part of the software safety program in the Software Safety Plan. (R.2950) – See NASA-STD-8719.13, 5.6.1

4.7.5.3 Operational Documentation

a. Ensure operational documents describe all safety related commands, data, input sequences, and options. (R.2960) – See NASA-STD-8719.13, 7.3

b. Ensure operational documents include error message descriptions and corrective actions. (R.2970) – See NASA-STD-8719.13, 7.3.1

4.7.5.4 Other Documentation Requirements

a. Address safety-critical software in all appropriate project documents. (R.2980) – See NASA-STD-8719.13, 5.6.3

4.8 SOW CONTENT REQUIREMENTS

4.8.1 Software Contract Requirements

a. Source code access **(R.2990)** – *See NPR 7150.2, SWE-042*

b. SW measurement data **(R.3000)** – *See NPR 7150.2, SWE-044*

c. Insight into test **(R.3010)** – *See NPR 7150.2, SWE-039*

d. Electronic access **(R.3020)** – *See NPR 7150.2, SWE-040*

e. Open source **(R.3030)** – *See NPR 7150.2, SWE-041*

f. Track change request **(R.3040)** – *See NPR 7150.2, SWE-043*

g. Joint audits **(R.3050)** – *See NPR 7150.2, SWE-045*

h. SW schedule **(R.3060)** – *See NPR 7150.2, SWE-046*

i. Traceability data **(R.3070)** – *See NPR 7150.2, SWE-047*

j. Solicitation **(R.3080)** – *See NPR 7150.2, SWE-048*

4.8.2 Software Safety Requirements

4.8.2.1 Contract Management

a. Contract/MOA/MOU requires safety-critical software be developed according to this Standard. (R.3090) – See NASA-STD-8719.13, 5.13.1

b. Software safety deliverables are included in the contract/MOA/MOU. (R.3100) – See NASA-STD-8719.13, 5.13.1.1

c. Customer surveillance for software safety is included in the contract/MOA/MOU. (R.3110) – See NASA-STD-8719.13, 5.13.1.2

d. The contract/MOA/MOU defines how the contractor and customer report and resolve software safety problems. (R.3120) – See NASA-STD-8719.13, 5.13.1.3

e. The contract/MOA/MOU defines that customer agreement is required for any changes to baselined safety-critical software elements. (R.3130) – See NASA-STD-8719.13, 5.13.1.4

4.9 METRICS REQUIREMENTS

4.9.1 Software Measurement

a. SW measurement areas **(R.3150)** – *See NPR 7150.2, SWE-091*

b. Collection and storage **(R.3160)** – *See NPR 7150.2, SWE-092*

c. Analyze data **(R.3170)** – *See NPR 7150.2, SWE-093*

d. Report analysis **(R.3180)** – *See NPR 7150.2, SWE-094*

e. Objectives **(R.3220)** – *See NPR 7150.2, SWE-090*

4.9.2 Software Report Requirements

a. SW Metrics Report **(R.3230)** – *See NPR 7150.2, SWE-117*

CHAPTER 5. COMPARISON TO NPR 7150.2

There is no direct correlation between the NASA-wide software classifications described in NPR 7150.2 and the classifications in this document. However, correlations have been made and agreed upon between the NASA software classifications and the previous Dryden software levels (A, B, and C). Specifically:

a. **Dryden Level A** (Software failure could cause loss of life, life-threatening injury, compromise public safety, or result in loss of or substantial damage to the vehicle/system/facility) is considered equivalent to NASA Class C, "Mission Support Software"

b. **Dryden Level B** (Software failure could cause loss of flight research mission/test) is considered equivalent to NASA Class D "Analysis and Distribution Software" and

c. **Dryden Level C** (Software failure could cause inaccurate results or inefficient use of resources) is considered equivalent to NASA Class E "Development Support Software"

Extending this comparison to the software classifications in this document,

a. DFRC Class I software is roughly equivalent to NASA Class C,

b. DFRC Class III is roughly equivalent to NASA Class D, and

c. DFRC Class IV is roughly equivalent to NASA Class E.

For the most part, this comparison holds. This chapter lists the areas where the Classifications in this document are either more stringent or less stringent than the classifications in NPR 7150.2, based on the equivalence listed above.

5.1 Areas where DPR 7150.2-001 is more stringent than NPR 7150.2

The following section describes those areas where the Dryden 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.

5.1.1 DFRC Class I requirements not included in NPR 7150.2 Class C

Table 3 documents the requirements required for DFRC Class I software that are not required for NASA Class C.

DPR 7150.2-001 Requirement ID	NPR 7150.2 ID Requirement ID	Description
R.0850	SWE-052	Bi-directional trace
R.0870	SWE-054	Corrective action
R.0920	SWE-058	Detailed design
R.0890	SWE-059	Bi-directional trace
R.0940	SWE-061	Coding standards
R.0930	SWE-064	Maintain Traceability
R.0970	SWE-070	Models, simulations, tools
R.1100	SWE-078	As-built documentation
R.2430	SWE-082	Authorizing Changes
R.2450	SWE-084	Perform configuration audits
R.1620	SWE-086	Continuous risk management
R.3140	SWE-089	Basic measurements
R.2630	SWE-105	SW Maintenance Plan
R.2640	SWE-106	SW Assurance Plan
R.2660	SWE-110	SW Data Dictionary
R.2710	SWE-115	SW Users Manual
R.2740	SWE-119	SW Inspection/Peer Review

Table 3 - DFRC Class I requirements not required for NASA Class C software

5.1.2 DFRC Class III requirements not included in NPR 7150.2 Level D

Table 4 documents the requirements required for DFRC Class III software that are not required for NASA Class D.

DPR 7150.2-001 Requirement ID	NPR 7150.2 ID Requirement ID	Description
R.0490	SWE-017	Training
R.0560	SWE-025	Corrective Action
R.0570	SWE-026	Changes
R.0580	SWE-027	COTS, GOTS, MOTS
R.0850	SWE-052	Bi-directional trace
R.0870	SWE-054	Corrective action
R.0880	SWE-055	Requirements Validation
R.0900	SWE-056	Document design
R.0920	SWE-058	Detailed design
R.1000	SWE-067	Test for compliance
R.1030	SWE-071	Update plans and procedures
R.1050	SWE-073	Platform or Hi-Fidelity simulation.
R.1060	SWE-074	Document maintenance plans
R.1070	SWE-075	Plan operations, maintenance, and retirement
R.1080	SWE-076	Implement plans
R.1100	SWE-078	As-built documentation
R.1110	SWE-087	Requirements and Test Plans
R.1120	SWE-088	Checklist, criteria, and tracking
R.1620	SWE-086	Continuous risk management
R.2430	SWE-082	Authorizing Changes
R.2440	SWE-083	Maintain records
R.2690	SWE-113	SW Change Request/ Problem Report
R.3040	SWE-043	Track change request
R.3050	SWE-045	Joint audits
R.3150	SWE-091	SW measurement areas
R.3160	SWE-092	Collection and storage

Table 4 - DFRC Class III requirements not required for NASA Class D software

5.1.3 DFRC Class IV requirements not included in NPR 7150.2 Level E

Table 5 documents the requirements required for DFRC Class IV software that are not required for NASA Class E.

DPR 7150.2-001 Requirement ID	NPR 7150.2 ID Requirement ID	Description
R.0570	SWE-026	Changes
R.0760	SWE-030	Verification results
R.0770	SWE-031	Validation results
R.0960	SWE-063	Version Description
R.1020	SWE-069	Document defect and track
R.2400	SWE-079	Develop CM plan
R.2410	SWE-080	Track and evaluate changes
R.2420	SWE-081	Identify S/W configuration items
R.2430	SWE-082	Authorizing Changes
R.2440	SWE-083	Maintain records
R.2720	SWE-116	SW Version Description

Table 5 - DFRC Class IV requirements not required for NASA Class E software

5.2 Areas where NPR 7150.2 is more stringent than DPR 7150.2

The following section describes those areas where the NASA requirements stated in NPR 7150.2 exceed the Dryden requirements for the corresponding classification of software. Note that requirements derived from other sources (NASA-STD-8739.13, DCP-S-007, or RTCA DO-178) are not listed. Note that these exemptions are not officially granted until a formal waiver is obtained from the NASA Office of the Chief Engineer.

5.2.1 NPR 7150.2 Level C requirements not included in DFRC Class I

Table 6 documents the requirements required for NASA Class C software that are not required for DFRC Class I.

DPR 7150.2-001 Requirement ID	NPR 7150.2 ID Requirement ID	Description
R.0590	SWE-032	CMM L3 or CMMI L2

Table 6 - NASA Class C software requirements not required for DFRC Class I software

5.2.2 NPR 7150.2 Level D requirements not included in DFRC Class III

Table 7 documents the requirements required for NASA Class D software that are not required for DFRC Class III.

DPR 7150.2-001 Requirement ID	NPR 7150.2 ID Requirement ID	Description
R.0470	SWE-015	Cost Estimation

Table 7 - NASA Class D software requirements not required for DFRC Class III software

5.2.3 NPR 7150.2 Level E requirements not included in DFRC Class IV

Table 8 documents the requirements required for NASA Class E software that are not required for DFRC Class IV.

DPR 7150.2-001 Requirement ID	NPR 7150.2 ID Requirement ID	Description
R.0470	SWE-015	Cost Estimation
R.0950	SWE-062	Unit test

Table 8 - NASA Class E software requirements not required for DFRC Class IV software

Appendix A Definitions

A.1 Coding standards. Written conventions specifying rules and guidelines for the proper use of an individual programming language's constructs, commenting, naming, and formatting, etc. The purpose of coding standards are to prevent programming errors, control complexity, and promote the understandability of the source code.

A.2 Commercial- Off-The-Shelf (COTS) Software. Operating systems, libraries, applications, and other software purchased from a commercial vendor. Not customized for a particular project. Access to source code and documentation are often limited.

A.3 Complex electronics. Complex electronics are programmable devices that can be used to implement specific hardware circuits. Examples include Complex Programmable Logic Devices (CPLDs), Field Programmable Gate Arrays (FPGAs), and Application Specific Integrated Circuits (ASICs).

A.4 Component level tests. Testing of the software at the functional component level.

A.5 CSCI .A unit of software that is treated as a single entity by the configuration management system.

A.6 Deactivated Code. Executable object code (or data) which by design is either

a) not intended to be executed (code) or used (data), for example, a part of a previously developed software component, or

b) is only executed (code) or used (data) in certain configurations of the target computer environment, for example, code that is enabled by a hardware pin selection or software programmed options.

A.7 Dead Code. Executable object code (or data) that, as a result of design error, cannot be executed (code) or used (data) in an operational configuration of the target computer environment and is not traceable to a system or software requirement. An exception is embedded identifiers.

A.8 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 aircraft operation.

A.9 Glueware Software. Created to connect the off-the-shelf software/reused software with the rest of the system. It may take the form of "adapters" that modify interfaces or add missing functionality, "firewalls" that isolate the off-the-shelf software, or "wrappers" that check inputs and outputs to the off-the-shelf software and may modify to prevent failures.

A.10 Government Off-The-Shelf (GOTS) Software. This refers to government-created software, usually from another project. The software was not created by the current developers (see software reuse). Usually, source code is included and all documentation, including test and analysis results, is available. That is, the government is responsible for the GOTS software to be incorporated into another system. (Definition from source document: NASA-GB-8719.13, NASA Software Safety Guidebook.)

A.11 Ground Software. Software that could indirectly impact flight or test operations. This includes software supporting simulation, control room, or data processing operations.

A.12 Heritage. See legacy.

A.13 Legacy. These are usually software products (architecture, code, requirements) written specifically for one project and then, without prior planning during its initial development, found to be useful on other projects.

A.14 Mission Critical. Item or function that must retain its operational capability to assure mission success. (Definition from source document: NPR 8715.3, NASA General Safety Program Requirements.)

A.15 Modified Off-The-Shelf (MOTS) Software. When COTS, legacy, reuse, or heritage software is changed to a certain degree, usually more than 10%, then it is considered "modified." The changes can include all or part of the software products and may involve additions, deletions, and specific alterations. An argument can be made that any alterations to the code and/or design of an off-the-shelf software component constitutes "modification"; however, the common usage allows for some percentage of change before the off-the-shelf software is declared to be MOTS software. This may include the changes to the application shell and/or glueware to add or protect against certain features and not to the off-the-shelf software system code directly.

A.16 Off-The-Shelf Software. Software not developed in-house or by a contractor for the specific project now underway. The software is general purpose or developed for a different purpose than the current project.

A.17 Partitioning. Separation, physically and/or logically, of safety-critical functions from other functionality.

A.18 Preliminary Hazard Analysis. A gross study of the initial system concepts. It is used to identify all of the energy sources that constitute inherent hazards. The energy sources are examined for possible accidents in every mode of system operation. The analysis is also used to identify methods of protection against all of the accident possibilities.

A.19 Reuse. See software reuse.

A.20 Risk Management. An organized, systematic decision-making process that efficiently identifies, analyzes, plans, tracks, controls, communicates, and documents risk to increase the likelihood of achieving program/project goals. (Definition from source document: NPR 8735.2, Management of Government Safety and Mission Assurance Surveillance Functions for NASA Contracts.)

A.21 Safety Critical. The term "safety critical" is as defined in NPR 8715.3, NASA General Safety Program Requirements.

A.22 Safety Critical Function. The term "safety critical function" is as defined in NPR 8715.3, NASA General Safety Program Requirements.

A.23 Software. Computer programs, procedures, rules, and associated documentation and data pertaining to the development and operation of a computer system. Software includes programs and operational data. This also includes COTS, GOTS, MOTS, reuse, auto code generated, firmware, and open source software components.

A.24 Software Engineering. The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software: that is, the application of engineering to software (Definition from source document: IEEE 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.)

A.25 Software Media. Device or material that acts as a means of transferral or storage of software, such as magnetic tape, CD, solid-state drive, or paper.

A.26 Software Reuse. A software product developed for one use but having other uses or one developed specifically to be usable on multiple projects or in multiple roles on one project. Examples include, but are not limited to, COTS products, acquirer-furnished software products, software products in reuse libraries, and pre-existing developer software products. Each use may include all or part of the software product and may involve its modification. This term can be applied to any software product (such as requirements and architectures), not just to software code itself. Often this is software previously written by an in-house development team and used on a different project. GOTS software would come under this category if it is supplied from one government project to another government project. (Definition from source document: NASA-GB-8719.13, NASA Software Safety Guidebook.)

A.27 Structural Coverage Analysis. Analysis to show which software structures were not covered by formal software testing

A.28 System. The combination of elements that function together to produce the capability required to meet a need. The elements include all hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose.

(Definition from source document: NPR 7120.5, NASA Program and Project Management Processes and Requirements.)

A.29 Traceability. Ability to trace the history, application, or location of an entity by means of recorded identifications. [ISO 8402, 3.16] For example, requirements traceability as applied to software safety involves identifying safety-critical requirements/functions then tracing them through design, test, acceptance, changes and upgrades, and through retirement

A.30 Unit level testing. Software testing based on internal structure of the unit under test that is done at or near the code level to ensure that the implementation matches the intended design.

A.31 Validation. Proof that the product accomplishes the intended purpose. May be determined by a combination of test, analysis, and demonstration. (Definition from source document: NPR 7120.5, NASA Program and Project Management Processes and Requirements.) Note: Software validation also includes software peer review and inspection.

A.32 Verification. Proof of compliance with specifications. May be determined by a combination of test, analysis, demonstration, and inspection. (Definitions from source document: NPR 7120.5, NASA Program and Project Management Processes and Requirements.)

Appendix B. Acronyms

ASIC	Application Specific Integrated Circuits
CDR	Critical Design Review
CIO	Chief Information Officer
CM	Configuration Management
COTS	Commercial-Off-The-Shelf
CPLD	Complex Programmable Logic Devices
CSCI	Computer Software Configuration Item
DCP	Dryden Centerwide Procedure
DGA	Designated Governing Authority
DPR	Dryden Procedural Requirement
FAR	Federal Acquisition Regulation
FPGA	Field Programmable Gate Arrays
GOTS	Government-Off-The-Shelf
IT	Information Technology
ITA	Independent Technical Authority
MOA	Memorandum of Agreement
MOTS	Modified-Off-The-Shelf
MOU	Memorandum of Understanding
NPR	NASA Procedural Requirement
ORR	Operational Readiness Review
OTS	Off-The-Shelf
PAG	Project Approval Group

PDR	Preliminary Design Review
PHA	Preliminary Hazard Analysis
RTCA	Radio Technical Commission for Aeronautics
SMA	Safety and Mission Assurance
SRR	System Requirements Review
SW	Software

Appendix C: Reference Documents

a. DCP-S-002, Hazard Management Procedure

b. DCP-S-007, Software Assurance

c. DPR 7123.1-001, Systems Engineering Requirements Document

d. NASA-STD-8719.13, NASA Software Safety Standard

e. NASA-STD-8739.8, Software Assurance Standard

f. NPR 2810.1, Security of Information Technology

g. NPR 7150.2, NASA Software Engineering Requirements

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

i. NPR 8715.3, NASA General Safety Program Requirements

j. RTCA DO-178, Software Considerations in Airborne Systems and Equipment Certification

Appendix D: Designated Governing Authority Allocations

D.1 SOFTWARE ENGINEERING REQUIREMENTS DELEGATED TO THE DRYDEN CENTER DIRECTOR

ID	Category	Requirement
R.0060	Applicability and Scope	Effective Date
R.0100	Best Practices	Identify applicable practices
R.0390	Compliance with Laws, Policies, and Requirements	SW Disclosures
R.0400	Compliance with Laws, Policies, and Requirements	Export Control
R.0410	Compliance with Laws, Policies, and Requirements	External Release
R.0430	Compliance with Laws, Policies, and Requirements	Disabilities - Reasonable Accommodation
R.0440	Compliance with Laws, Policies, and Requirements	Disabilities - Section 508 Compliance
R.0170	Expertise of ITA Warrant Authority(s)	Non-IT & Non Business
R.0180	Expertise of ITA Warrant Authority(s)	IT infrastructure & Business
R.0070	Organizational Capability and Improvement	Center Plan
R.0080	Organizational Capability and Improvement	SW Processes
R.0150	Tailoring of Requirements	Alternate requirement request
R.0160	Tailoring of Requirements	Document approved alternate requirements
R.0110	Training	Software engineering training

D.2 SOFTWARE ENGINEERING REQUIREMENTS DELEGATED TO THE ASSOCIATE DIRECTOR FOR MISSION SUPPORT, DIRECTOR FOR RESEARCH AND ENGINEERING, DIRECTOR FOR MISSION SYSTEMS, DIRECTOR FOR FLIGHT OPERATIONS, AND THE SAFETY AND MISSION ASSURANCE LEAD

The Software Engineering Requirements listed in the following table are delegated to each of the individuals listed above for software development and procurement activities within their respective organizations:

ID	Category	Requirement
R.0580	Commercial, Government, and Modified Off-The-Shelf Software	COTS, GOTS, MOTS
R.0190	Compliance	Direction for Warrant Authority
R.2750	Compliance	Compliance Matrix
R.0200	Compliance	Considerations for Waivers
R.0210	Compliance	Review of "P(Center)"
R.0220	Compliance	Compliance Records
R.1110	Peer Reviews/Inspections	Requirements & Test Plans
R.1120	Peer Reviews/Inspections	Checklist, criteria, & tracking
R.3140	Peer Reviews/Inspections	Basic measurements
R.2740	Peer Reviews/Inspections	SW Inspection/Peer Review
R.0780	Project Formulation Requirements	Options for Acquisition
R.0790	Project Formulation Requirements	Acceptance Criteria
R.0600	Project Formulation Requirements	Supplier Selection
R.0800	Project Formulation Requirements	SW processes & tasks
R.0810	Project Formulation Requirements	Milestones
R.0610	Project Formulation Requirements	Acquisition planning
R.2400	Software Configuration Management	Develop CM plan
R.2410	Software Configuration Management	Track & evaluate changes
R.2420	Software Configuration Management	Identify S/W configuration items
R.2430	Software Configuration Management	Authorizing Changes

ID	Category	Requirement
R.2440	Software Configuration Management	Maintain records
R.2450	Software Configuration Management	Perform configuration audits
R.2460	Software Configuration Management	Implement procedures
R.0922	Software Design	Low Level Algorithm Review
R.0900	Software Design	Document design
R.0910	Software Design	Architecture
R.0920	Software Design	Detailed design
R.0890	Software Design	Bi-directional trace
R.2650	Software Design Documents	SW Requirements Spec.
R.2660	Software Design Documents	SW Data Dictionary
R.2670	Software Design Documents	SW Design Description
R.2680	Software Design Documents	Interface Design Description
R.2690	Software Design Documents	SW Change Request/ Problem Report
R.2700	Software Design Documents	SW Test Procedures
R.2710	Software Design Documents	SW Users Manual
R.2720	Software Design Documents	SW Version Description
R.1580	Software Implementation	Design --> code
R.0940	Software Implementation	Coding standards
R.0950	Software Implementation	Unit test
R.0960	Software Implementation	Version Description
R.0930	Software Implementation	Maintain Traceability
R.0450	Software Life Cycle Planning	SW Plan
R.0460	Software Life Cycle Planning	Execute Plan
R.0470	Software Life Cycle Planning	Cost Estimation
R.0480	Software Life Cycle Planning	Schedule
R.0490	Software Life Cycle Planning	Training

ID	Category	Requirement
R.0500	Software Life Cycle Planning	Reviews
R.0510	Software Life Cycle Planning	Life Cycle
R.0520	Software Life Cycle Planning	SW Classification
R.0530	Software Life Cycle Planning	SW Classification changes
R.0550	Software Life Cycle Planning	Plan Tracking
R.0560	Software Life Cycle Planning	Corrective Action
R.0570	Software Life Cycle Planning	Changes
R.3220	Software Measurement	Objectives
R.3150	Software Measurement	SW measurement areas
R.3160	Software Measurement	Collection & storage
R.3170	Software Measurement	Analyze data
R.3180	Software Measurement	Report analysis
R.1060	Software Operations, Maintenance, and Retirement	Document maintenance. plans
R.1070	Software Operations, Maintenance, and Retirement	Plan operations., maintenance. & retirement
R.1080	Software Operations, Maintenance, and Retirement	Implement plans
R.1090	Software Operations, Maintenance, and Retirement	Deliver software product
R.1100	Software Operations, Maintenance, and Retirement	As-built documentation
R.2610	Software Plans	SW Configuration Mgt. Plan
R.2640	Software Plans	SW Assurance Plan
R.2600	Software Plans	SW Development/Mgt. Plan
R.2630	Software Plans	SW Maintenance Plan
R.2620	Software Plans	SW Test Plan
R.3230	Software Report Requirements	SW Metrics Report

ID	Category	Requirement
R.2730	Software Reports	SW Test Report
R.0882	Software Requirements	High Level Algorithm Review
R.0924	Software Requirements	Software Architecture Review vs. high level Requirements
R.0926	Software Requirements	Review of Software Architecture data and control flow
R.0928	Software Requirements	Review of Software Architecture partitioning
R.0820	Software Requirements	Documented Requirements
R.0830	Software Requirements	SW requirements
R.0840	Software Requirements	Flow-down & derived req.
R.0850	Software Requirements	Bi-directional trace
R.0860	Software Requirements	Manage req. change
R.0870	Software Requirements	Corrective action
R.0880	Software Requirements	Requirements Validation
R.1002	Software Testing	Performance of Structural Coverage Analysis
R.1004	Software Testing	Utilize Structural Coverage Analysis to identify additional testing requirements
R.1006	Software Testing	Dead Code Identification and Removal
R.1008	Software Testing	Deactivated Code Analysis
R.0980	Software Testing	Plan, procedures, reports
R.0990	Software Testing	Perform testing
R.1000	Software Testing	Test for compliance
R.1010	Software Testing	Evaluate test results
R.1020	Software Testing	Doc. defect & track
R.0970	Software Testing	Models, simulations, tools
R.1030	Software Testing	Update plans & procedures
R.1040	Software Testing	Maintain Traceability

ID	Category	Requirement
R.1050	Software Testing	Platform or Hi-Fidelity simulation.
R.0740	Software Verification and Validation	Verification planning
R.0750	Software Verification and Validation	Validation planning
R.0760	Software Verification and Validation	Verification results
R.0770	Software Verification and Validation	Validation results

D.3 SOFTWARE ENGINEERING REQUIREMENTS DELEGATED TO THE DIRECTOR FOR RESEARCH AND ENGINEERING

ID	Category	Requirement
R.0130	Software Plans	Center SW Training Plan
R.0140	Software Plans	Center SW Engineering Improve Plan
R.0120	Training	Software training plan

D.4 SOFTWARE ENGINEERING REQUIREMENTS DELEGATED TO THE DIRECTOR FOR MISSION SYSTEMS

ID	Category	Requirement
R.0420	Compliance with Laws, Policies, and Requirements	Security

D.5 SOFTWARE ENGINEERING REQUIREMENTS DELEGATED TO THE DIRECTOR FOR FLIGHT OPERATIONS

ID	Category	Requirement
R.2461	Software Configuration Management	Labeling of Software Media
R.2462	Software Configuration Management	Version Description Document
R.2463	Software Configuration Management	Flight Media Release Form included in Version Description Document
R.2464	Software Configuration Management	Flight Software Installation Procedure
R.2465	Software Configuration Management	Specification of flight(s) on Flight Media Release Form

ID	Category	Requirement
R.2466	Software Configuration Management	Confirmation of Correct Software Version before flight

D.6 SOFTWARE ENGINEERING REQUIREMENTS DELEGATED TO THE DIRECTOR FOR SAFETY AND MISSION ASSURANCE

ID	Category	Requirement
R.0090	Compliance with Laws, Policies, and Requirements	SW Safety
R.1620	Risk Management	Continuous risk management
R.0540	Software Life Cycle Planning	SW Assurance
R.0240	Software Safety Requirements	Establish a certification process for safety-critical software. Safety-critical software is certified prior to use or release.
R.0250	Software Safety Requirements	Participate in program/project/facility certifications
R.0260	Software Safety Requirements	5.14.3-(a-g) are items to be evaluated for certification
R.0270	Software Safety Requirements	All software hazards are identified.
R.0280	Software Safety Requirements	All hazard controls that require software implementation are identified.
R.0290	Software Safety Requirements	All software safety requirements and elements are identified and tracked.
R.0300	Software Safety Requirements	All software safety requirements and elements have been successfully validated, or waivers/deviations have been approved.
R.0310	Software Safety Requirements	All software safety requirements and elements have been properly verified, or waivers/deviations have been approved.
R.0320	Software Safety Requirements	All discrepancies in safety-critical software have been dispositioned with the safety organization's concurrence, per the certification process.
R.0330	Software Safety Requirements	All operational workarounds associated with discrepancies in safety-critical software have the concurrence of the Center or Program safety organization, per the certification process.

ID	Category	Requirement
R.1930	Software Safety Requirements	Present the software safety process and results to an appropriate safety panel for certification.
R.0340	Software Safety Requirements	Approve the results and reports prior to acceptance of the software and the system, with review and certification provided by SMA .
R.3090	Software Safety Requirements	Contract/MOA/MOU requires safety-critical software be developed according to this Standard.
R.3100	Software Safety Requirements	Software safety deliverables are included in the contract/MOA/MOU.
R.3110	Software Safety Requirements	Customer surveillance for software safety is included in the contract/MOA/MOU.
R.3120	Software Safety Requirements	The contract/MOA/MOU defines how the contractor and customer report and resolve software safety problems.
R.3130	Software Safety Requirements	The contract/MOA/MOU defines that customer agreement is required for any changes to baselined safety-critical software elements.
R.0230	Software Safety Requirements	SMA approves of the evaluation conclusions.
R.1130	Software Safety Requirements	If the system is safety-critical, evaluate the software.
R.1140	Software Safety Requirements	Assume all software is safety critical until proven otherwise.
R.1150	Software Safety Requirements	Use the criteria in this section to determine if the software is safety-critical
R.1160	Software Safety Requirements	Evaluate software during project planning
R.1170	Software Safety Requirements	Document the results of the evaluation
R.1180	Software Safety Requirements	Evaluate all software in the system
R.2480	Software Safety Requirements	Create closed-loop tracking of discrepancies, problems, and failures
R.2490	Software Safety Requirements	Trace identified safety-critical software problems back to the system-level hazard involved.
R.2500	Software Safety Requirements	Approve safety-critical discrepancy report closures.

ID	Category	Requirement
R.2510	Software Safety Requirements	Regularly review all discrepancy reports for safety impacts.
R.2520	Software Safety Requirements	Evaluate software changes for potential safety impact.
R.0730	Software Safety Requirements	Define the change and approval process for software safety related portions of all project documents.
R.1210	Software Safety Requirements	Evaluate all off-the-shelf and reused software for its impact safety-critical functions.
R.1220	Software Safety Requirements	Analyze safety-critical OTS and reused software for its ability to meet required safety functions, any safety impact of extra functionality, and interfaces to developed code.
R.1230	Software Safety Requirements	Analyze the interactions of COTS software components with the developed software and any other COTS software in the system.
R.1240	Software Safety Requirements	Verify safety-critical OTS or reused software to the same level required of in-house developed software to the extent possible.
R.2960	Software Safety Requirements	Ensure operational documents describe all safety related commands, data, input sequences, and options.
R.2970	Software Safety Requirements	Ensure operational documents include error message descriptions and corrective actions.
R.2350	Software Safety Requirements	Present the results of the software safety design analysis at project formal reviews and system-level safety reviews.
R.0350	Software Safety Requirements	This Standard applies to safety-critical software that has been released for operations.
R.1570	Software Safety Requirements	When operational software is changed, specify, develop, analyze, and test all safety-critical software.
R.2360	Software Safety Requirements	Evaluate proposed changes for their impact on system safety.
R.2370	Software Safety Requirements	Assess the amount of regression testing needed.
R.2380	Software Safety Requirements	Concur on any changes to basic, as built, or approved upgrades of the operational software.

ID	Category	Requirement
R.2390	Software Safety Requirements	Review updates to user manuals and procedures for safety impacts, and verify software safety related operational workarounds are properly documented.
R.0360	Software Safety Requirements	When a system or facility is retired, this Standard no longer applies.
R.0370	Software Safety Requirements	A retirement plan will address the safe termination of operations, decommissioning, and retirement of the system or facility.
R.2980	Software Safety Requirements	Address safety-critical software in all appropriate project documents.
R.0720	Software Safety Requirements	Assign software assurance to verify that software safety is planned, approved, and implemented.
R.2470	Software Safety Requirements	Perform configuration change control, status accounting, and change verification of safety-critical software requirements and software elements.
R.1190	Software Safety Requirements	Include software safety in project planning
R.0620	Software Safety Requirements	Consult with software safety personnel when acquiring safety-critical software
R.0630	Software Safety Requirements	Periodically evaluate the system for safety-critical software
R.0640	Software Safety Requirements	Provides adequate resources to the software safety program.
R.0650	Software Safety Requirements	Assign personnel for the software safety program
R.0660	Software Safety Requirements	Work with SMA management to resolve conflicts
R.0670	Software Safety Requirements	Plan and execute software safety throughout the entire software life cycle.
R.0680	Software Safety Requirements	Integrate software safety with system safety and software development.
R.0690	Software Safety Requirements	Create a process to document, trace, communicate, and close software safety concerns
R.1630	Software Safety Requirements	Participate in system safety analyses
R.1640	Software Safety Requirements	Evaluate hazards for software's contribution (cause, control, etc.)

ID	Category	Requirement
R.1650	Software Safety Requirements	Conduct software safety analyses; coordinate with the system safety analyses.
R.1660	Software Safety Requirements	Create software safety requirements
R.1670	Software Safety Requirements	Document software safety requirements, hazard contributors and controls. Software safety plan points to this document.
R.1890	Software Safety Requirements	Complete all assurance activities prior to acceptance or closure of any software-related system-level hazards.
R.1900	Software Safety Requirements	Coordinate software safety tasks with software assurance.
R.2530	Software Safety Requirements	Manage configuration of software, documentation, and associated data
R.2540	Software Safety Requirements	Maintain all baselined safety-critical software and associated documentation, simulators, models, test suites, data, etc.
R.2550	Software Safety Requirements	Provide and document the release of safety-critical software.
R.2560	Software Safety Requirements	Evaluate all changes, modifications, and patches made to safety-critical requirements, design, code, systems, equipment, test plans, procedures, simulators, models, test suites, or criteria.
R.2570	Software Safety Requirements	Approve changes to baselined safety-critical software.
R.2580	Software Safety Requirements	Track and control incremental changes to the safety-critical software and its release to operations.
R.2590	Software Safety Requirements	Maintain configuration control of routine reconfigurations and changes to operational software
R.1330	Software Safety Requirements	Incorporate all functional software safety requirements into the software design.
R.1340	Software Safety Requirements	Identify safety design features and methods in the software design.
R.1350	Software Safety Requirements	Allow for software safety features and requirements be thoroughly tested in the software design.
R.1360	Software Safety Requirements	Designate design elements that implement safety-critical requirements as safety-critical.

ID	Category	Requirement
R.1370	Software Safety Requirements	Clearly identify all safety-critical design elements in software design documentation.
R.1380	Software Safety Requirements	Modularize the safety-related aspects of the design in the software design.
R.2040	Software Safety Requirements	Analyze the software design.
R.2050	Software Safety Requirements	Document the analysis methodology
R.2060	Software Safety Requirements	Ensure documented analysis method or procedure meets the following requirements (a-f):
R.2070	Software Safety Requirements	Verify software design meets the requirements of section 6.2.1 and subsections.
R.2080	Software Safety Requirements	Verify design does not compromise any safety controls or processes. Ensure that any additional hazard cause or contribution is documented. The design maintains the system in a safe state during all modes of operation.
R.2090	Software Safety Requirements	Verify safety features are adequate for their function.
R.2100	Software Safety Requirements	Determine design features necessary to prevent, mitigate or control failures and faults, and the partitioning of safety features between hardware and software.
R.2110	Software Safety Requirements	Verify that any partitioning or isolation methods adequately isolate the safety-critical design elements from those that are non-safety-critical.
R.2120	Software Safety Requirements	Verify all safety-critical design elements are traceable to software safety requirements, and vice versa.
R.2130	Software Safety Requirements	Provide the documented results of the analysis and any newly identified hazards to system safety.
R.2140	Software Safety Requirements	Include the results of the software safety design analysis at project formal reviews and system-level safety reviews.
R.2770	Software Safety Requirements	Define approach to preventing software tools from introducing hazards
R.2780	Software Safety Requirements	Identify & assess project tools that could potentially impact safety-critical software and define mitigation strategies if necessary.

ID	Category	Requirement
R.2790	Software Safety Requirements	Document how project tools are selected, approved, and controlled.
R.2800	Software Safety Requirements	Document how approved tools are upgraded, what happens if an approved tool is no longer approved, and what limitations are imposed on tool use.
R.2760	Software Safety Requirements	Document the resource requirements and allocation for software safety tasks.
R.1590	Software Safety Requirements	Implement all software safety design features and methods in the software code.
R.1600	Software Safety Requirements	Incorporate software coding standards that prohibit unsafe language features and require commenting of safety-critical source code.
R.1610	Software Safety Requirements	Utilize software coding standard in developing code.
R.2150	Software Safety Requirements	Analyze the software implementation (e.g., code).
R.2160	Software Safety Requirements	Document the analysis methodology
R.2170	Software Safety Requirements	Ensure the analysis method or procedure meets the following requirements (a-e) and is documented
R.2180	Software Safety Requirements	Verify safety-critical software code and data meets the requirements of section 6.3.1 and subsections.
R.2190	Software Safety Requirements	Verify design safety features and methods are correctly implemented in the software code.
R.2200	Software Safety Requirements	Verify the software maintains the system in a safe state during all modes of operation and does not compromise any safety controls or processes, nor create any additional hazards.
R.2210	Software Safety Requirements	Ensure code and data verification activities include software safety requirements, if a requirement can be verified at this level.
R.2220	Software Safety Requirements	Verify all safety-critical code units are traceable to safety-critical design elements.
R.2230	Software Safety Requirements	Provide the documented results of the analysis and any newly identified hazards to system safety.

ID	Category	Requirement
R.2240	Software Safety Requirements	Include the results of the software safety design analysis at project formal reviews and system-level safety reviews.
R.1390	Software Safety Requirements	Verify unit testing and data verification is completed before the unit is integrated.
R.1830	Software Safety Requirements	Perform software safety activities throughout the software development life cycle.
R.1840	Software Safety Requirements	Continue software safety activities in the operational phase.
R.0700	Software Safety Requirements	Assign S/W safety manager to develop and implement S/W safety processes and plans
R.1680	Software Safety Requirements	Communicate software safety concerns directly to the project manager
R.1690	Software Safety Requirements	Elevate software safety concerns that cannot be resolved within the project.
R.1700	Software Safety Requirements	Assure software safety risks are captured, addressed, and managed as part of risk management processes
R.1710	Software Safety Requirements	Participate in change control board for software modifications
R.1720	Software Safety Requirements	Brief management on the selection of off-the-shelf or previously created (reused) software used in this system
R.1730	Software Safety Requirements	Provide inputs to management regarding contractor requirements for safety-critical software.
R.0710	Software Safety Requirements	Assign personnel to perform software safety activities
R.1740	Software Safety Requirements	Analyze and report software safety non-conformances to appropriate personnel
R.1750	Software Safety Requirements	Review system hazard analyses for changes that impact the software subsystem.
R.1760	Software Safety Requirements	Inform system safety personnel of changes in safety-critical software
R.1770	Software Safety Requirements	Support the system safety review process.
R.1780	Software Safety Requirements	Participate in project reviews.
R.2810	Software Safety Requirements	Cross-reference safety activities that are in multiple plans.

ID	Category	Requirement
R.2820	Software Safety Requirements	Put the Software Safety Plan under configuration control.
R.2830	Software Safety Requirements	Describe how the requirements of this Standard will be implemented .
R.2840	Software Safety Requirements	Include activities, schedule, personnel, methods, and resulting products.
R.2850	Software Safety Requirements	Define how system safety, software assurance, software development, and the Center or Program SMA organization works together.
R.2860	Software Safety Requirements	Describe the role of IV&V and how IV&V will work with the software safety program and personnel.
R.2870	Software Safety Requirements	Describe how safety-critical requirements are generated, implemented, tracked, and verified.
R.2880	Software Safety Requirements	Define the procedures for resolving software safety concerns and recommendations.
R.2890	Software Safety Requirements	Describe how software safety and project schedules are synchronized.
R.2900	Software Safety Requirements	Specify the number and schedule of software safety assurance audits.
R.2910	Software Safety Requirements	Document the conditions requiring software safety engineers to review a situation and proposed solutions or upgrades.
R.2920	Software Safety Requirements	Define who monitors system during operation, & what procedures are followed when they feel safety may be threatened.
R.2930	Software Safety Requirements	Describe the training requirements for all project software safety roles.
R.2940	Software Safety Requirements	Describe how software safety is integrated with the chosen software life cycle.
R.2950	Software Safety Requirements	List the documents (and associated content) that are part of the software safety program in the Software Safety Plan.
R.1790	Software Safety Requirements	Perform software safety assessment and planning for each software acquisition, maintenance activity, or change to legacy systems.
R.1200	Software Safety Requirements	Plan and conduct safety program reviews

ID	Category	Requirement
R.1800	Software Safety Requirements	Start software safety planning early enough to affect the software development and assurance activities.
R.1810	Software Safety Requirements	Create a Software Safety Plan.
R.1820	Software Safety Requirements	Review the Software Safety Plan periodically.
R.1270	Software Safety Requirements	Create software safety requirements and include them in the software requirements specification.
R.1280	Software Safety Requirements	Derive software safety requirements from system safety, environmental, interface, vehicle, and facility requirements; standards, program specification, system hazard reports and analyses
R.1290	Software Safety Requirements	Clearly identify software safety requirements in the software requirements specification.
R.1300	Software Safety Requirements	Express and structure software safety requirements that are clear, precise, unequivocal, verifiable, testable, maintainable and feasible.
R.1310	Software Safety Requirements	Include the modes or states of operation under which software safety requirements are valid or not applicable.
R.1320	Software Safety Requirements	Include hardware and software safety-related constraints in software requirements document.
R.1940	Software Safety Requirements	Analyze software safety requirements.
R.1950	Software Safety Requirements	Ensure the analysis method or procedure meets the following requirements (a-e) and is documented:
R.1960	Software Safety Requirements	Verify all software safety requirements meet the requirements of section 6.1.1 and subsections
R.1970	Software Safety Requirements	Examine the software safety requirements for ambiguities, inconsistencies, omissions, and undefined conditions
R.1980	Software Safety Requirements	Verify all software safety requirements are traceable to higher-level requirements or external standards
R.1990	Software Safety Requirements	Verify software safety requirements provide adequate response to potential failures.

ID	Category	Requirement
R.2000	Software Safety Requirements	Verify software safety requirements include positive measures to prevent potential problems and implement required “must work” functions.
R.2010	Software Safety Requirements	Document results of the analysis. Give any newly identified hazards to system safety.
R.2020	Software Safety Requirements	Document project-level resolution for improperly decomposed requirements.
R.2030	Software Safety Requirements	Include results of the software safety requirements analysis at project formal reviews and system-level safety reviews.
R.1400	Software Safety Requirements	Verify all functional software safety requirements and safety-critical software elements by testing.
R.1410	Software Safety Requirements	Verify via test that system hazards related to software have been eliminated or controlled to an acceptable level of risk.
R.1420	Software Safety Requirements	Include software safety testing in unit level tests and component level tests.
R.1430	Software Safety Requirements	Configuration control unit test simulators, test drivers, stubs, and test data.
R.1440	Software Safety Requirements	Configuration control component test simulators, test drivers, stubs, and test data.
R.1450	Software Safety Requirements	Document the results of unit level and component level tests, plus the test procedures, simulators, test suites, drivers, stubs and data used.
R.1460	Software Safety Requirements	Include software safety testing within system and acceptance tests.
R.1470	Software Safety Requirements	Verify the correct and safe operation of the software in conjunction with system hardware and operator inputs prior to system acceptance.
R.1480	Software Safety Requirements	Verify the correct and safe operation of the system in the presence of failures and faults.
R.1490	Software Safety Requirements	Use safety analyses to determine which failures to test for, how many, and in what combinations.
R.1500	Software Safety Requirements	Verify the correct and safe operation of the system under system load, stress, and off-nominal conditions.

ID	Category	Requirement
R.1510	Software Safety Requirements	Verify the system operates correctly and safely in all anticipated operational and off-nominal configurations.
R.1520	Software Safety Requirements	Analyze newly identified hazardous states or contributors prior to software delivery or use.
R.1530	Software Safety Requirements	Evaluate, inspect, or demonstrate requirements if testing is not feasible.
R.1540	Software Safety Requirements	Record the rationale for choosing evaluation, inspection, or demonstration over test.
R.1550	Software Safety Requirements	Record the evaluation, inspection, or demonstration methodology.
R.1560	Software Safety Requirements	Get the software safety engineer's concurrence with both the rationale and the methodology.
R.2250	Software Safety Requirements	Analyze results from the software and system test process or requirements verification process.
R.2260	Software Safety Requirements	Document the analysis methodology
R.2270	Software Safety Requirements	Ensure the analysis method or procedure meets the following requirements (a-d) and is documented.
R.2280	Software Safety Requirements	Verify software and system tests data meet the requirements of section 6.4.1 and subsections.
R.2290	Software Safety Requirements	Verify that the requirements verification evaluation, inspection, or demonstration data meet the requirements of section 6.4.2 and subsections.
R.2300	Software Safety Requirements	Verify test coverage analysis shows that all safety requirements, functions, controls, and processes have been completely covered.
R.2310	Software Safety Requirements	Verify software safety requirements have been tested, or evaluated, inspected, or demonstrated.
R.2320	Software Safety Requirements	Verify all software safety functions are correctly performed and the software system does not perform unintended functions.
R.2330	Software Safety Requirements	Document the results of the analysis and provide any newly identified hazards to system safety.

ID	Category	Requirement
R.2340	Software Safety Requirements	Document and report improperly implemented requirements for project-level resolution.
R.1910	Software Safety Requirements	Ensure sufficient safety testing and analysis is performed when a project tool changes
R.1920	Software Safety Requirements	Approve the safety analysis and testing approach for tool verification.
R.1850	Software Safety Requirements	Create a tracing system that maps software safety requirements to system hazards and traces the flow down of to design, implementation, and test.
R.1860	Software Safety Requirements	Coordinate the software tracing system with the system-level hazard tracking system.
R.1870	Software Safety Requirements	Put the tracing system under configuration control.
R.1880	Software Safety Requirements	Review the tracing system reports and outputs.
R.1250	Software Safety Requirements	Request a waiver/deviation if a requirement cannot be met.
R.1260	Software Safety Requirements	Document in a written request the justification to support the waiver/deviation.
R.0380	Software Safety Requirements	Keep copies of all variances to safety requirements.

D.7 SOFTWARE ENGINEERING REQUIREMENTS DELEGATED TO THE ACQUISITION MANAGEMENT OFFICER

ID	Category	Requirement
R.3010	Software Contract Requirements	Insight into test
R.3020	Software Contract Requirements	Electronic access
R.3030	Software Contract Requirements	Open source
R.2990	Software Contract Requirements	Source code access
R.3040	Software Contract Requirements	Track change request
R.3000	Software Contract Requirements	SW measurement data
R.3050	Software Contract Requirements	Joint audits

ID	Category	Requirement
R.3060	Software Contract Requirements	SW schedule
R.3070	Software Contract Requirements	Traceability data
R.3080	Software Contract Requirements	Solicitation

THIS PAGE INTENTIONALY LEFT BLANK

Appendix E: Requirements Mapping Matrix

The following tables map the requirements specified in this document with the corresponding software classification. The classification definitions are provided in Chapter 2. Note that the "S" classification (Safety Critical) is applied in addition to the numeric classification. In other words, software classified as III-S would be subject to the requirements in both the "III" column and the "S" column.

E.1 CENTER LEVEL SOFTWARE ENGINEERING REQUIREMENTS

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Applicability and Scope	**R.0060**	Effective Date	NPR 7150.2, SWE-001						
Organizational Capability and Improvement	**R.0070**	Center Plan	NPR 7150.2, SWE-003						
Organizational Capability and Improvement	**R.0080**	SW Processes	NPR 7150.2, SWE-005						
Best Practices	**R.0100**	Identify applicable practices	NPR 7150.2, SWE-099						
Training	**R.0110**	Software engineering training	NPR 7150.2, SWE-100						
Training	**R.0120**	Software training plan	NPR 7150.2, SWE-101						
Software Plans	**R.0130**	Center SW Training Plan	NPR 7150.2, SWE-107						
Software Plans	**R.0140**	Center SW Engineering Improve Plan	NPR 7150.2, SWE-108						
Tailoring of Requirements	**R.0150**	Alternate requirement request	NPR 7150.2, SWE-120						
Tailoring of Requirements	**R.0160**	Document approved alternate requirements	NPR 7150.2, SWE-121						
Expertise of ITA Warrant Authority(s)	**R.0170**	Non-IT and Non-Business	NPR 7150.2, SWE-122						
Expertise of ITA Warrant Authority(s)	**R.0180**	IT infrastructure and Business	NPR 7150.2, SWE-123						
Compliance	**R.0190**	Direction for Warrant Authority	NPR 7150.2, SWE-124						
Compliance	**R.0200**	Considerations for Waivers	NPR 7150.2, SWE-126						

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Compliance	**R.0210**	Review of "P(Center)"	NPR 7150.2, SWE-127						This standard will define center requirements.
Compliance	**R.0220**	Compliance Records	NPR 7150.2, SWE-128						
Software Safety Requirements	**R.0230**	SMA approves of the evaluation conclusions.	NASA-STD-8719.13, 4.1.1.5					**x**	
Software Safety Requirements	**R.0240**	Establish a certification process for safety-critical software. Safety-critical software is certified prior to use or release.	NASA-STD-8719.13, 5.14.1					**x**	
Software Safety Requirements	**R.0250**	Participate in program/project/facility certifications	NASA-STD-8719.13, 5.14.2					**x**	
Software Safety Requirements	**R.0260**	5.14.3-(a-g) are items to be evaluated for certification	NASA-STD-8719.13, 5.14.3					**x**	
Software Safety Requirements	**R.0270**	All software hazards are identified.	NASA-STD-8719.13, 5.14.3-a					**x**	
Software Safety Requirements	**R.0280**	All hazard controls that require software implementation are identified.	NASA-STD-8719.13, 5.14.3-b					**x**	
Software Safety Requirements	**R.0290**	All software safety requirements and elements are identified and tracked.	NASA-STD-8719.13, 5.14.3-c					**x**	
Software Safety Requirements	**R.0300**	All software safety requirements and elements have been successfully validated, or waivers/deviations have been approved.	NASA-STD-8719.13, 5.14.3-d					**x**	
Software Safety Requirements	**R.0310**	All software safety requirements and elements have been properly verified, or waivers/deviations have been approved.	NASA-STD-8719.13, 5.14.3-e					**x**	
Software Safety Requirements	**R.0320**	All discrepancies in safety-critical software have been dispositioned with the safety organization's concurrence, per the certification process.	NASA-STD-8719.13, 5.14.3-f					**x**	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.0330**	All operational workarounds associated with discrepancies in safety-critical software have the concurrence of the Center or Program safety organization, per the certification process.	NASA-STD-8719.13, 5.14.3-g					x	
Software Safety Requirements	**R.0340**	Approve the results and reports prior to acceptance of the software and the system, with review and certification provided by SMA .	NASA-STD-8719.13, 5.14.5					x	
Software Safety Requirements	**R.0350**	This Standard applies to safety-critical software that has been released for operations.	NASA-STD-8719.13, 7.1					x	
Software Safety Requirements	**R.0360**	When a system or facility is retired, this Standard no longer applies.	NASA-STD-8719.13, 7.4					x	
Software Safety Requirements	**R.0370**	A retirement plan will address the safe termination of operations, decommissioning, and retirement of the system or facility.	NASA-STD-8719.13, 7.4.1					x	
Software Safety Requirements	**R.0380**	Keep copies of all variances to safety requirements.	NASA-STD-8719.13, 5.15.3					x	

E.2 PROJECT LEVEL SOFTWARE ENGINEERING REQUIREMENTS

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Compliance with Laws, Policies, and Requirements	**R.0090**	SW Safety	NPR 7150.2, SWE-023	x	x	x	x		NASA-STD-8719.13 requirements already folded in to this standard.
Compliance with Laws, Policies, and Requirements	**R.0390**	SW Disclosures	NPR 7150.2, SWE-007	x	x	x	x		
Compliance with Laws, Policies, and Requirements	**R.0400**	Export Control	NPR 7150.2, SWE-008	x	x	x	x		
Compliance with Laws, Policies, and Requirements	**R.0410**	External Release	NPR 7150.2, SWE-009	x	x	x	x		
Compliance with Laws, Policies, and Requirements	**R.0420**	Security	NPR 7150.2, SWE-010	x	x	x	x		
Compliance with Laws, Policies, and Requirements	**R.0430**	Disabilities - Reasonable Accommodation	NPR 7150.2, SWE-011	x	x	x	x		
Compliance with Laws, Policies, and Requirements	**R.0440**	Disabilities - Section 508 Compliance	NPR 7150.2, SWE-012	x	x	x	x		
Software Life Cycle Planning	**R.0450**	SW Plan	NPR 7150.2, SWE-013	x	x	x	x		
Software Life Cycle Planning	**R.0460**	Execute Plan	NPR 7150.2, SWE-014	x	x	x	x		
Software Life Cycle Planning	**R.0470**	Cost Estimation	NPR 7150.2, SWE-015	x	x				
Software Life Cycle Planning	**R.0480**	Schedule	NPR 7150.2, SWE-016	x	x	x			
Software Life Cycle Planning	**R.0490**	Training	NPR 7150.2, SWE-017	x	x	x			

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Life Cycle Planning	**R.0500**	Reviews	NPR 7150.2, SWE-018	x	x	x			
Software Life Cycle Planning	**R.0510**	Life Cycle	NPR 7150.2, SWE-019	x	x	x			
Software Life Cycle Planning	**R.0520**	SW Classification	NPR 7150.2, SWE-020	x	x	x	x		
Software Life Cycle Planning	**R.0530**	SW Classification changes	NPR 7150.2, SWE-021	x	x	x	x		
Software Life Cycle Planning	**R.0540**	SW Assurance	NPR 7150.2, SWE-022	x	x				
Software Life Cycle Planning	**R.0550**	Plan Tracking	NPR 7150.2, SWE-024	x	x	x			
Software Life Cycle Planning	**R.0560**	Corrective Action	NPR 7150.2, SWE-025	x	x	x			
Software Life Cycle Planning	**R.0570**	Changes	NPR 7150.2, SWE-026	x	x	x	x		
Commercial, Government, and Modified Off-The-Shelf Software	**R.0580**	COTS, GOTS, MOTS	NPR 7150.2, SWE-027	x	x	x			
Project Formulation Requirements	**R.0600**	Supplier Selection	NPR 7150.2, SWE-035	x	x				
Project Formulation Requirements	**R.0610**	Acquisition planning	NPR 7150.2, SWE-038	x	x	x			
Software Safety Requirements	**R.0620**	Consult with software safety personnel when acquiring safety-critical software	NASA-STD-8719.13, 5.1.2.1.1					x	
Software Safety Requirements	**R.0630**	Periodically evaluate the system for safety-critical software	NASA-STD-8719.13, 5.1.2.1.2					x	
Software Safety Requirements	**R.0640**	Provides adequate resources to the software safety program.	NASA-STD-8719.13, 5.1.2.1.3					x	
Software Safety Requirements	**R.0650**	Assign personnel for the software safety program	NASA-STD-8719.13, 5.1.2.1.4					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.0660**	Work with SMA management to resolve conflicts	NASA-STD-8719.13, 5.1.2.1.5					x	
Software Safety Requirements	**R.0670**	Plan and execute software safety throughout the entire software life cycle.	NASA-STD-8719.13, 5.1.2.2					x	
Software Safety Requirements	**R.0680**	Integrate software safety with system safety and software development.	NASA-STD-8719.13, 5.1.2.3					x	
Software Safety Requirements	**R.0690**	Create a process to document, trace, communicate, and close software safety concerns	NASA-STD-8719.13, 5.1.2.4					x	
Software Safety Requirements	**R.0700**	Assign S/W safety manager to develop and implement S/W safety processes and plans	NASA-STD-8719.13, 5.1.3.1					x	
Software Safety Requirements	**R.0710**	Assign personnel to perform software safety activities	NASA-STD-8719.13, 5.1.3.2					x	
Software Safety Requirements	**R.0720**	Assign software assurance to verify that software safety is planned, approved, and implemented.	NASA-STD-8719.13, 5.1.4.1					x	
Software Safety Requirements	**R.0730**	Define the change and approval process for software safety related portions of all project documents.	NASA-STD-8719.13, 5.6.2					x	

E.3 SYSTEM LEVEL SOFTWARE ENGINEERING REQUIREMENTS

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Verification and Validation	**R.0740**	Verification planning	NPR 7150.2, SWE-028	x	x	x			
Software Verification and Validation	**R.0750**	Validation planning	NPR 7150.2, SWE-029	x	x	x			
Software Verification and Validation	**R.0760**	Verification results	NPR 7150.2, SWE-030	x	x	x	x		
Software Verification and Validation	**R.0770**	Validation results	NPR 7150.2, SWE-031	x	x	x	x		
Project Formulation Requirements	**R.0780**	Options for Acquisition	NPR 7150.2, SWE-033	x	x	x			
Project Formulation Requirements	**R.0790**	Acceptance Criteria	NPR 7150.2, SWE-034	x	x	x			
Project Formulation Requirements	**R.0800**	SW processes and tasks	NPR 7150.2, SWE-036	x	x	x			
Project Formulation Requirements	**R.0810**	Milestones	NPR 7150.2, SWE-037	x	x	x			
Software Requirements	**R.0820**	Documented Requirements	NPR 7150.2, SWE-049	x	x	x	x		
Software Requirements	**R.0830**	SW requirements	NPR 7150.2, SWE-050	x	x	x			
Software Requirements	**R.0840**	Flow-down and derived req.	NPR 7150.2, SWE-051	x	x				
Software Requirements	**R.0850**	Bi-directional trace	NPR 7150.2, SWE-052	x	x	x			

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Requirements	**R.0860**	Manage req. change	NPR 7150.2, SWE-053	x	x	x	x		
Software Requirements	**R.0870**	Corrective action	NPR 7150.2, SWE-054	x	x	x			
Software Requirements	**R.0880**	Requirements Validation	NPR 7150.2, SWE-055	x(i)	x	x			
Software Requirements	**R.0882**	High Level Algorithm Review	DO-178, 6.3.1g	x(i)	x				
Software Requirements	**R.0924**	Software Architecture Review vs. high level Requirements	DO-178, 6.3.3a	x(i)	x				
Software Requirements	**R.0926**	Review of Software Architecture data and control flow	DO-178, 6.3.3b	x(i)	x				
Software Requirements	**R.0928**	Review of Software Architecture partitioning	DO-178, 6.3.3c	x(i)	x				
Software Design	**R.0890**	Bi-directional trace	NPR 7150.2, SWE-059	x	x				
Software Design	**R.0900**	Document design	NPR 7150.2, SWE-056	x	x	x			
Software Design	**R.0910**	Architecture	NPR 7150.2, SWE-057	x	x	x			
Software Design	**R.0920**	Detailed design	NPR 7150.2, SWE-058	x	x	x			
Software Design	**R.0922**	Low Level Algorithm Review	DO-178, 6.3.2g	x(i)	x				
Software Implementation	**R.0930**	Maintain Traceability	NPR 7150.2, SWE-064	x	x				
Software Implementation	**R.0940**	Coding standards	NPR 7150.2, SWE-061	x(i)	x				
Software Implementation	**R.0950**	Unit test	NPR 7150.2, SWE-062	x	x	x			
Software Implementation	**R.0960**	Version Description	NPR 7150.2, SWE-063	x	x	x	x		
Software Testing	**R.0970**	Models, simulations, tools	NPR 7150.2, SWE-070	x	x				
Software Testing	**R.0980**	Plan, procedures, reports	NPR 7150.2, SWE-065	x	x	x			
Software Testing	**R.0990**	Perform testing	NPR 7150.2, SWE-066	x	x	x			
Software Testing	**R.1000**	Test for compliance	NPR 7150.2, SWE-067	x(i)	x	x			
Software Testing	**R.1002**	Performance of Structural Coverage Analysis	DO-178, 6.4.4.2	x(i)	x				

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Testing	**R.1004**	Utilize Structural Coverage Analysis to identify additional testing requirements	DO-178, 6.4.4.3	x(i)	x				
Software Testing	**R.1006**	Dead Code Identification and Removal	DO-178, 6.4.4.3c	x(i)	x				
Software Testing	**R.1008**	Deactivated Code Analysis	DO-178, 6.4.4.3d	x(i)	x				
Software Testing	**R.1010**	Evaluate test results	NPR 7150.2, SWE-068	x(i)	x	x			
Software Testing	**R.1020**	Document defect and track	NPR 7150.2, SWE-069	x	x	x	x		
Software Testing	**R.1030**	Update plans and procedures	NPR 7150.2, SWE-071	x	x	x			
Software Testing	**R.1040**	Maintain Traceability	NPR 7150.2, SWE-072	x	x	x			
Software Testing	**R.1050**	Platform or Hi-Fidelity simulation.	NPR 7150.2, SWE-073	x	x	x			
Software Operations, Maintenance, and Retirement	**R.1060**	Document maintenance. plans	NPR 7150.2, SWE-074	x	x	x			
Software Operations, Maintenance, and Retirement	**R.1070**	Plan operations, maintenance, and retirement	NPR 7150.2, SWE-075	x	x	x			
Software Operations, Maintenance, and Retirement	**R.1080**	Implement plans	NPR 7150.2, SWE-076	x	x	x			
Software Operations, Maintenance, and Retirement	**R.1090**	Deliver software product	NPR 7150.2, SWE-077	x	x	x	x		
Software Operations, Maintenance, and Retirement	**R.1100**	As-built documentation	NPR 7150.2, SWE-078	x	x	x			
Peer Reviews/Inspections	**R.1110**	Requirements and Test Plans	NPR 7150.2, SWE-087	x	x	x			

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Peer Reviews/Inspections	**R.1120**	Checklist, criteria, and tracking	NPR 7150.2, SWE-088	x	x	x			
Peer Reviews/Inspections	**R.2740**	SW Inspection/Peer Review	NPR 7150.2, SWE-119	x	x				
Peer Reviews/Inspections	**R.3140**	Basic measurements	NPR 7150.2, SWE-089	x	x				
Software Safety Requirements	**R.1130**	If the system is safety-critical, evaluate the software.	NASA-STD-8719.13, 4.1.1					x	
Software Safety Requirements	**R.1140**	Assume all software is safety critical until proven otherwise.	NASA-STD-8719.13, 4.1.1.1					x	
Software Safety Requirements	**R.1150**	Use the criteria in this section to determine if the software is safety-critical	NASA-STD-8719.13, 4.1.1.2					x	
Software Safety Requirements	**R.1160**	Evaluate software during project planning	NASA-STD-8719.13, 4.1.1.3					x	
Software Safety Requirements	**R.1170**	Document the results of the evaluation	NASA-STD-8719.13, 4.1.1.4					x	
Software Safety Requirements	**R.1180**	Evaluate all software in the system	NASA-STD-8719.13, 4.1.2					x	
Software Safety Requirements	**R.1190**	Include software safety in project planning	NASA-STD-8719.13, 5.1.2.1					x	
Software Safety Requirements	**R.1200**	Plan and conduct safety program reviews	NASA-STD-8719.13, 5.2.1.1					x	
Software Safety Requirements	**R.1210**	Evaluate all off-the-shelf and reused software for its impact safety-critical functions.	NASA-STD-8719.13, 5.12.1					x	
Software Safety Requirements	**R.1220**	Analyze safety-critical OTS and reused software for its ability to meet required safety functions, any safety impact of extra functionality, and interfaces to developed code.	NASA-STD-8719.13, 5.12.1.1					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.1230**	Analyze the interactions of COTS software components with the developed software and any other COTS software in the system.	NASA-STD-8719.13, 5.12.1.2					x	
Software Safety Requirements	**R.1240**	Verify safety-critical OTS or reused software to the same level required of in-house developed software to the extent possible.	NASA-STD-8719.13, 5.12.1.3					x	
Software Safety Requirements	**R.1250**	Request a waiver/deviation if a requirement cannot be met.	NASA-STD-8719.13, 5.15.1					x	
Software Safety Requirements	**R.1260**	Document in a written request the justification to support the waiver/deviation.	NASA-STD-8719.13, 5.15.2					x	
Software Safety Requirements	**R.1270**	Create software safety requirements and include them in the software requirements specification.	NASA-STD-8719.13, 6.1.1					x	
Software Safety Requirements	**R.1280**	Derive software safety requirements from system safety, environmental, interface, vehicle, and facility requirements; standards, program specification, system hazard reports and analyses	NASA-STD-8719.13, 6.1.1.1					x	
Software Safety Requirements	**R.1290**	Clearly identify software safety requirements in the software requirements specification.	NASA-STD-8719.13, 6.1.1.2					x	
Software Safety Requirements	**R.1300**	Express and structure software safety requirements that are clear, precise, unequivocal, verifiable, testable, maintainable, and feasible.	NASA-STD-8719.13, 6.1.1.3					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.1310**	Include the modes or states of operation under which software safety requirements are valid or not applicable.	NASA-STD-8719.13, 6.1.1.4					x	
Software Safety Requirements	**R.1320**	Include hardware and software safety-related constraints in software requirements document.	NASA-STD-8719.13, 6.1.1.5					x	
Software Safety Requirements	**R.1330**	Incorporate all functional software safety requirements into the software design.	NASA-STD-8719.13, 6.2.1					x	
Software Safety Requirements	**R.1340**	Identify safety design features and methods in the software design.	NASA-STD-8719.13, 6.2.1.1					x	
Software Safety Requirements	**R.1350**	Allow for software safety features and requirements be thoroughly tested in the software design.	NASA-STD-8719.13, 6.2.1.2					x	
Software Safety Requirements	**R.1360**	Designate design elements that implement safety-critical requirements as safety-critical.	NASA-STD-8719.13, 6.2.1.3					x	
Software Safety Requirements	**R.1370**	Clearly identify all safety-critical design elements in software design documentation.	NASA-STD-8719.13, 6.2.1.3.1					x	
Software Safety Requirements	**R.1380**	Modularize the safety-related aspects of the design in the software design.	NASA-STD-8719.13, 6.2.1.4					x	
Software Safety Requirements	**R.1390**	Verify unit testing and data verification is completed before the unit is integrated.	NASA-STD-8719.13, 6.3.3					x	
Software Safety Requirements	**R.1400**	Verify all functional software safety requirements and safety-critical software elements by testing.	NASA-STD-8719.13, 6.4.1					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.1410**	Verify via test that system hazards related to software have been eliminated or controlled to an acceptable level of risk.	NASA-STD-8719.13, 6.4.1.1					x	
Software Safety Requirements	**R.1420**	Include software safety testing in unit level tests and component level tests.	NASA-STD-8719.13, 6.4.1.2					x	
Software Safety Requirements	**R.1430**	Configuration control unit test simulators, test drivers, stubs, and test data.	NASA-STD-8719.13, 6.4.1.2.1					x	
Software Safety Requirements	**R.1440**	Configuration control component test simulators, test drivers, stubs, and test data.	NASA-STD-8719.13, 6.4.1.2.2					x	
Software Safety Requirements	**R.1450**	Document the results of unit level and component level tests, plus the test procedures, simulators, test suites, drivers, stubs, and data used.	NASA-STD-8719.13, 6.4.1.2.3					x	
Software Safety Requirements	**R.1460**	Include software safety testing within system and acceptance tests.	NASA-STD-8719.13, 6.4.1.3					x	
Software Safety Requirements	**R.1470**	Verify the correct and safe operation of the software in conjunction with system hardware and operator inputs prior to system acceptance.	NASA-STD-8719.13, 6.4.1.3.1					x	
Software Safety Requirements	**R.1480**	Verify the correct and safe operation of the system in the presence of failures and faults.	NASA-STD-8719.13, 6.4.1.3.2					x	
Software Safety Requirements	**R.1490**	Use safety analyses to determine which failures to test for, how many, and in what combinations.	NASA-STD-8719.13, 6.4.1.3.3					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.1500**	Verify the correct and safe operation of the system under system load, stress, and off-nominal conditions.	NASA-STD-8719.13, 6.4.1.3.4					x	
Software Safety Requirements	**R.1510**	Verify the system operates correctly and safely in all anticipated operational and off-nominal configurations.	NASA-STD-8719.13, 6.4.1.3.5					x	
Software Safety Requirements	**R.1520**	Analyze newly identified hazardous states or contributors prior to software delivery or use.	NASA-STD-8719.13, 6.4.1.4					x	
Software Safety Requirements	**R.1530**	Evaluate, inspect, or demonstrate requirements if testing is not feasible.	NASA-STD-8719.13, 6.4.2					x	
Software Safety Requirements	**R.1540**	Record the rationale for choosing evaluation, inspection, or demonstration over test.	NASA-STD-8719.13, 6.4.2.1					x	
Software Safety Requirements	**R.1550**	Record the evaluation, inspection, or demonstration methodology.	NASA-STD-8719.13, 6.4.2.2					x	
Software Safety Requirements	**R.1560**	Get the software safety engineer's concurrence with both the rationale and the methodology.	NASA-STD-8719.13, 6.4.2.3					x	
Software Safety Requirements	**R.1570**	When operational software is changed, specify, develop, analyze, and test all safety-critical software.	NASA-STD-8719.13, 7.2					x	

E.4 DEVELOPER LEVEL SOFTWARE ENGINEERING REQUIREMENTS

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Implementation	**R.1580**	Design --> code	NPR 7150.2, SWE-060	x	x	x			
Software Safety Requirements	**R.1590**	Implement all software safety design features and methods in the software code.	NASA-STD-8719.13, 6.3.1					x	
Software Safety Requirements	**R.1600**	Incorporate software coding standards that prohibit unsafe language features and require commenting of safety-critical source code.	NASA-STD-8719.13, 6.3.1.1					x	
Software Safety Requirements	**R.1610**	Utilize software coding standard in developing code.	NASA-STD-8719.13, 6.3.1.2					x	

E.5 SYSTEM SAFETY SOFTWARE ENGINEERING REQUIREMENTS

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Risk Management	**R.1620**	Continuous risk management	NPR 7150.2, SWE-086	x	x	x			
Software Safety Requirements	**R.1630**	Participate in system safety analyses	NASA-STD-8719.13, 4.2.1					x	
Software Safety Requirements	**R.1640**	Evaluate hazards for software's contribution (cause, control, etc.)	NASA-STD-8719.13, 4.2.1.1					x	
Software Safety Requirements	**R.1650**	Conduct software safety analyses; coordinate with the system safety analyses.	NASA-STD-8719.13, 4.2.1.2					x	
Software Safety Requirements	**R.1660**	Create software safety requirements	NASA-STD-8719.13, 4.2.2					x	
Software Safety Requirements	**R.1670**	Document software safety requirements, hazard contributors, and controls. Software safety plan points to this document.	NASA-STD-8719.13, 4.2.2.1					x	
Software Safety Requirements	**R.1680**	Communicate software safety concerns directly to the project manager	NASA-STD-8719.13, 5.1.3.1.1					x	
Software Safety Requirements	**R.1690**	Elevate software safety concerns that cannot be resolved within the project.	NASA-STD-8719.13, 5.1.3.1.2					x	
Software Safety Requirements	**R.1700**	Assure software safety risks are captured, addressed, and managed as part of risk management processes	NASA-STD-8719.13, 5.1.3.1.3					x	
Software Safety Requirements	**R.1710**	Participate in change control board for software modifications	NASA-STD-8719.13, 5.1.3.1.4					x	
Software Safety Requirements	**R.1720**	Brief management on the selection of off-the-shelf or previously created (reused) software used in this system	NASA-STD-8719.13, 5.1.3.1.5					x	
Software Safety Requirements	**R.1730**	Provide inputs to management regarding contractor requirements for safety-critical software.	NASA-STD-8719.13, 5.1.3.1.6					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.1740**	Analyze and report software safety nonconformances to appropriate personnel	NASA-STD-8719.13, 5.1.3.2.1					x	
Software Safety Requirements	**R.1750**	Review system hazard analyses for changes that impact the software subsystem.	NASA-STD-8719.13, 5.1.3.2.2					x	
Software Safety Requirements	**R.1760**	Inform system safety personnel of changes in safety-critical software	NASA-STD-8719.13, 5.1.3.2.3					x	
Software Safety Requirements	**R.1770**	Support the system safety review process.	NASA-STD-8719.13, 5.1.3.2.4					x	
Software Safety Requirements	**R.1780**	Participate in project reviews.	NASA-STD-8719.13, 5.1.3.2.5					x	
Software Safety Requirements	**R.1790**	Perform software safety assessment and planning for each software acquisition, maintenance activity, or change to legacy systems.	NASA-STD-8719.13, 5.2.1					x	
Software Safety Requirements	**R.1800**	Start software safety planning early enough to affect the software development and assurance activities.	NASA-STD-8719.13, 5.2.2					x	
Software Safety Requirements	**R.1810**	Create a Software Safety Plan.	NASA-STD-8719.13, 5.2.3					x	
Software Safety Requirements	**R.1820**	Review the Software Safety Plan periodically.	NASA-STD-8719.13, 5.2.7					x	
Software Safety Requirements	**R.1830**	Perform software safety activities throughout the software development life cycle.	NASA-STD-8719.13, 5.5.2					x	
Software Safety Requirements	**R.1840**	Continue software safety activities in the operational phase.	NASA-STD-8719.13, 5.5.3					x	
Software Safety Requirements	**R.1850**	Create a tracing system that maps software safety requirements to system hazards and traces the flow down of to design, implementation, and test.	NASA-STD-8719.13, 5.7.1					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.1860**	Coordinate the software tracing system with the system-level hazard tracking system.	NASA-STD-8719.13, 5.7.1.1					x	
Software Safety Requirements	**R.1870**	Put the tracing system under configuration control.	NASA-STD-8719.13, 5.7.2					x	
Software Safety Requirements	**R.1880**	Review the tracing system reports and outputs.	NASA-STD-8719.13, 5.7.3					x	
Software Safety Requirements	**R.1890**	Complete all assurance activities prior to acceptance or closure of any software-related system-level hazards.	NASA-STD-8719.13, 5.10.1					x	
Software Safety Requirements	**R.1900**	Coordinate software safety tasks with software assurance.	NASA-STD-8719.13, 5.10.2					x	
Software Safety Requirements	**R.1910**	Ensure sufficient safety testing and analysis is performed when a project tool changes	NASA-STD-8719.13, 5.11.1.2.2					x	
Software Safety Requirements	**R.1920**	Approve the safety analysis and testing approach for tool verification.	NASA-STD-8719.13, 5.11.2					x	
Software Safety Requirements	**R.1930**	Present the software safety process and results to an appropriate safety panel for certification.	NASA-STD-8719.13, 5.14.4					x	
Software Safety Requirements	**R.1940**	Analyze software safety requirements.	NASA-STD-8719.13, 6.1.2					x	
Software Safety Requirements	**R.1950**	Ensure the analysis method or procedure meets the following requirements (a-e) and is documented:	NASA-STD-8719.13, 6.1.2.1					x	
Software Safety Requirements	**R.1960**	Verify all software safety requirements meet the requirements of section 6.1.1 and subsections	NASA-STD-8719.13, 6.1.2.1-a					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.1970**	Examine the software safety requirements for ambiguities, inconsistencies, omissions, and undefined conditions	NASA-STD-8719.13, 6.1.2.1-b					x	
Software Safety Requirements	**R.1980**	Verify all software safety requirements are traceable to higher-level requirements or external standards	NASA-STD-8719.13, 6.1.2.1-c					x	
Software Safety Requirements	**R.1990**	Verify software safety requirements provide adequate response to potential failures.	NASA-STD-8719.13, 6.1.2.1-d					x	
Software Safety Requirements	**R.2000**	Verify software safety requirements include positive measures to prevent potential problems and implement required “must work” functions.	NASA-STD-8719.13, 6.1.2.1-e					x	
Software Safety Requirements	**R.2010**	Document results of the analysis. Give any newly identified hazards to system safety.	NASA-STD-8719.13, 6.1.2.2					x	
Software Safety Requirements	**R.2020**	Document project-level resolution for improperly decomposed requirements.	NASA-STD-8719.13, 6.1.2.3					x	
Software Safety Requirements	**R.2030**	Include results of the software safety requirements analysis at project formal reviews and system-level safety reviews.	NASA-STD-8719.13, 6.1.2.4					x	
Software Safety Requirements	**R.2040**	Analyze the software design.	NASA-STD-8719.13, 6.2.2					x	
Software Safety Requirements	**R.2050**	Document the analysis methodology	NASA-STD-8719.13, 6.2.2.1					x	
Software Safety Requirements	**R.2060**	Ensure documented analysis method or procedure meets the following requirements (a-f):	NASA-STD-8719.13, 6.2.2.2					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.2070**	Verify software design meets the requirements of section 6.2.1 and subsections.	NASA-STD-8719.13, 6.2.2.2-a					x	
Software Safety Requirements	**R.2080**	Verify design does not compromise any safety controls or processes. Ensure that any additional hazard cause or contribution is documented. The design maintains the system in a safe state during all modes of operation.	NASA-STD-8719.13, 6.2.2.2-b					x	
Software Safety Requirements	**R.2090**	Verify safety features are adequate for their function.	NASA-STD-8719.13, 6.2.2.2-c					x	
Software Safety Requirements	**R.2100**	Determine design features necessary to prevent, mitigate, or control failures and faults, and the partitioning of safety features between hardware and software.	NASA-STD-8719.13, 6.2.2.2-d					x	
Software Safety Requirements	**R.2110**	Verify that any partitioning or isolation methods adequately isolate the safety-critical design elements from those that are non-safety-critical.	NASA-STD-8719.13, 6.2.2.2-e					x	
Software Safety Requirements	**R.2120**	Verify all safety-critical design elements are traceable to software safety requirements, and vice versa.	NASA-STD-8719.13, 6.2.2.2-f					x	
Software Safety Requirements	**R.2130**	Provide the documented results of the analysis and any newly identified hazards to system safety.	NASA-STD-8719.13, 6.2.2.3					x	
Software Safety Requirements	**R.2140**	Include the results of the software safety design analysis at project formal reviews and system-level safety reviews.	NASA-STD-8719.13, 6.2.2.4					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.2150**	Analyze the software implementation (e.g., code).	NASA-STD-8719.13, 6.3.2					x	
Software Safety Requirements	**R.2160**	Document the analysis methodology	NASA-STD-8719.13, 6.3.2.1					x	
Software Safety Requirements	**R.2170**	Ensure the analysis method or procedure meets the following requirements (a-e) and is documented	NASA-STD-8719.13, 6.3.2.2					x	
Software Safety Requirements	**R.2180**	Verify safety-critical software code and data meets the requirements of section 6.3.1 and subsections.	NASA-STD-8719.13, 6.3.2.2-a					x	
Software Safety Requirements	**R.2190**	Verify design safety features and methods are correctly implemented in the software code.	NASA-STD-8719.13, 6.3.2.2-b					x	
Software Safety Requirements	**R.2200**	Verify the software maintains the system in a safe state during all modes of operation and does not compromise any safety controls or processes, nor create any additional hazards.	NASA-STD-8719.13, 6.3.2.2-c					x	
Software Safety Requirements	**R.2210**	Ensure code and data verification activities include software safety requirements, if a requirement can be verified at this level.	NASA-STD-8719.13, 6.3.2.2-d					x	
Software Safety Requirements	**R.2220**	Verify all safety-critical code units are traceable to safety-critical design elements.	NASA-STD-8719.13, 6.3.2.2-e					x	
Software Safety Requirements	**R.2230**	Provide the documented results of the analysis and any newly identified hazards to system safety.	NASA-STD-8719.13, 6.3.2.3					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.2240**	Include the results of the software safety design analysis at project formal reviews and system-level safety reviews.	NASA-STD-8719.13, 6.3.2.4					x	
Software Safety Requirements	**R.2250**	Analyze results from the software and system test process or requirements verification process.	NASA-STD-8719.13, 6.4.3					x	
Software Safety Requirements	**R.2260**	Document the analysis methodology	NASA-STD-8719.13, 6.4.3.1					x	
Software Safety Requirements	**R.2270**	Ensure the analysis method or procedure meets the following requirements (a-d) and is documented.	NASA-STD-8719.13, 6.4.3.2					x	
Software Safety Requirements	**R.2280**	Verify software and system tests data meet the requirements of section 6.4.1 and subsections.	NASA-STD-8719.13, 6.4.3.2-a					x	
Software Safety Requirements	**R.2290**	Verify that the requirements verification evaluation, inspection, or demonstration data meet the requirements of section 6.4.2 and subsections.	NASA-STD-8719.13, 6.4.3.2-b					x	
Software Safety Requirements	**R.2300**	Verify test coverage analysis shows that all safety requirements, functions, controls, and processes have been completely covered.	NASA-STD-8719.13, 6.4.3.2-c					x	
Software Safety Requirements	**R.2310**	Verify software safety requirements have been tested, or evaluated, inspected, or demonstrated.	NASA-STD-8719.13, 6.4.3.2-d					x	
Software Safety Requirements	**R.2320**	Verify all software safety functions are correctly performed and the software system does not perform unintended functions.	NASA-STD-8719.13, 6.4.3.2-e					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.2330**	Document the results of the analysis and provide any newly identified hazards to system safety.	NASA-STD-8719.13, 6.4.3.3					x	
Software Safety Requirements	**R.2340**	Document and report improperly implemented requirements for project-level resolution.	NASA-STD-8719.13, 6.4.3.4					x	
Software Safety Requirements	**R.2350**	Present the results of the software safety design analysis at project formal reviews and system-level safety reviews.	NASA-STD-8719.13, 6.4.3.5					x	
Software Safety Requirements	**R.2360**	Evaluate proposed changes for their impact on system safety.	NASA-STD-8719.13, 7.2.1					x	
Software Safety Requirements	**R.2370**	Assess the amount of regression testing needed.	NASA-STD-8719.13, 7.2.1.1					x	
Software Safety Requirements	**R.2380**	Concur on any changes to basic, as built, or approved upgrades of the operational software.	NASA-STD-8719.13, 7.2.1.2					x	
Software Safety Requirements	**R.2390**	Review updates to user manuals and procedures for safety impacts, and verify software safety related operational workarounds are properly documented.	NASA-STD-8719.13, 7.3.2					x	

E.6 CONFIGURATION MANAGEMENT REQUIREMENTS

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Configuration Management	**R.2400**	Develop CM plan	NPR 7150.2, SWE-079	x	x	x	x		
Software Configuration Management	**R.2410**	Track and evaluate changes	NPR 7150.2, SWE-080	x	x	x	x		
Software Configuration Management	**R.2420**	Identify S/W configuration items	NPR 7150.2, SWE-081	x	x	x	x		
Software Configuration Management	**R.2430**	Authorizing Changes	NPR 7150.2, SWE-082	x	x	x	x		
Software Configuration Management	**R.2440**	Maintain records	NPR 7150.2, SWE-083	x	x	x	x		
Software Configuration Management	**R.2450**	Perform configuration audits	NPR 7150.2, SWE-084	x(i)	x				
Software Configuration Management	**R.2460**	Implement procedures	NPR 7150.2, SWE-085	x	x	x	x		
Software Configuration Management	**R.2461**	Labeling of Software Media	DCP-S-007, p27	x	x	x	x		flight software only
Software Configuration Management	**R.2462**	Version Description Document	DCP-S-007, p27	x	x	x	x		flight software only
Software Configuration Management	**R.2463**	Flight Media Release Form included in Version Description Document	DCP-S-007, p27	x	x	x	x		flight software only
Software Configuration Management	**R.2464**	Flight Software Installation Procedure	DCP-S-007, p28	x	x	x	x		flight software only

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Configuration Management	**R.2465**	Specification of flight(s) on Flight Media Release Form	DCP-S-007, p28	x	x	x	x		flight software only
Software Configuration Management	**R.2466**	Confirmation of Correct Software Version before flight	DCP-S-007, p28	x	x	x	x		flight software only
Software Safety Requirements	**R.2470**	Perform configuration change control, status accounting, and change verification of safety-critical software requirements and software elements.	NASA-STD-8719.13, 5.1.4.2					x	
Software Safety Requirements	**R.2480**	Create closed-loop tracking of discrepancies, problems, and failures	NASA-STD-8719.13, 5.8.1					x	
Software Safety Requirements	**R.2490**	Trace identified safety-critical software problems back to the system-level hazard involved.	NASA-STD-8719.13, 5.8.1.1					x	
Software Safety Requirements	**R.2500**	Approve safety-critical discrepancy report closures.	NASA-STD-8719.13, 5.8.1.2					x	
Software Safety Requirements	**R.2510**	Regularly review all discrepancy reports for safety impacts.	NASA-STD-8719.13, 5.8.2					x	
Software Safety Requirements	**R.2520**	Evaluate software changes for potential safety impact.	NASA-STD-8719.13, 5.8.3					x	
Software Safety Requirements	**R.2530**	Manage configuration of software, documentation, and associated data	NASA-STD-8719.13, 5.9.1					x	
Software Safety Requirements	**R.2540**	Maintain all baselined safety-critical software and associated documentation, simulators, models, test suites, data, etc.	NASA-STD-8719.13, 5.9.1.1					x	
Software Safety Requirements	**R.2550**	Provide and document the release of safety-critical software.	NASA-STD-8719.13, 5.9.1.2					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.2560**	Evaluate all changes, modifications, and patches made to safety-critical requirements, design, code, systems, equipment, test plans, procedures, simulators, models, test suites, or criteria.	NASA-STD-8719.13, 5.9.2					x	
Software Safety Requirements	**R.2570**	Approve changes to baselined safety-critical software.	NASA-STD-8719.13, 5.9.2.1					x	
Software Safety Requirements	**R.2580**	Track and control incremental changes to the safety-critical software and its release to operations.	NASA-STD-8719.13, 5.9.3					x	
Software Safety Requirements	**R.2590**	Maintain configuration control of routine reconfigurations and changes to operational software.	NASA-STD-8719.13, 5.9.3.1					x	

E.7 DOCUMENTATION REQUIREMENTS

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Plans	**R.2610**	SW Configuration Mgt. Plan	NPR 7150.2, SWE-103	x	x	x			
Software Plans	**R.2640**	SW Assurance Plan	NPR 7150.2, SWE-106	x	x				
Software Plans	**R.2600**	SW Development/Mgt. Plan	NPR 7150.2, SWE-102	x	x	x			
Software Plans	**R.2630**	SW Maintenance Plan	NPR 7150.2, SWE-105	x	x				
Software Plans	**R.2620**	SW Test Plan	NPR 7150.2, SWE-104	x	x	x			
Software Design Documents	**R.2650**	SW Requirements Spec.	NPR 7150.2, SWE-109	x	x	x			
Software Design Documents	**R.2660**	SW Data Dictionary	NPR 7150.2, SWE-110	x	x				
Software Design Documents	**R.2670**	SW Design Description	NPR 7150.2, SWE-111	x	x	x			
Software Design Documents	**R.2680**	Interface Design Description	NPR 7150.2, SWE-112	x	x				
Software Design Documents	**R.2690**	SW Change Request/ Problem Report	NPR 7150.2, SWE-113	x	x	x			
Software Design Documents	**R.2700**	SW Test Procedures	NPR 7150.2, SWE-114	x	x				
Software Design Documents	**R.2710**	SW Users Manual	NPR 7150.2, SWE-115	x	x				
Software Design Documents	**R.2720**	SW Version Description	NPR 7150.2, SWE-116	x	x	x	x		
Software Reports	**R.2730**	SW Test Report	NPR 7150.2, SWE-118	x	x				
Compliance	**R.2750**	Compliance Matrix	NPR 7150.2, SWE-125	x	x	x	x		
Software Safety Requirements	**R.2760**	Document the resource requirements and allocation for software safety tasks.	NASA-STD-8719.13, 5.4.1					x	
Software Safety Requirements	**R.2770**	Define approach to preventing software tools from introducing hazards	NASA-STD-8719.13, 5.11.1					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.2780**	Identify and assess project tools that could potentially impact safety-critical software and define mitigation strategies if necessary.	NASA-STD-8719.13, 5.11.1.1					x	
Software Safety Requirements	**R.2790**	Document how project tools are selected, approved, and controlled.	NASA-STD-8719.13, 5.11.1.2					x	
Software Safety Requirements	**R.2800**	Document how approved tools are upgraded, what happens if an approved tool is no longer approved, and what limitations are imposed on tool use.	NASA-STD-8719.13, 5.11.1.2.1					x	
Software Safety Requirements	**R.2810**	Cross-reference safety activities that are in multiple plans.	NASA-STD-8719.13, 5.2.3.1					x	
Software Safety Requirements	**R.2820**	Put the Software Safety Plan under configuration control.	NASA-STD-8719.13, 5.2.3.2					x	
Software Safety Requirements	**R.2830**	Describe how the requirements of this Standard will be implemented.	NASA-STD-8719.13, 5.2.4					x	
Software Safety Requirements	**R.2840**	Include activities, schedule, personnel, methods, and resulting products.	NASA-STD-8719.13, 5.2.5					x	
Software Safety Requirements	**R.2850**	Define how system safety, software assurance, software development, and the Center or Program SMA organization works together.	NASA-STD-8719.13, 5.2.6					x	
Software Safety Requirements	**R.2860**	Describe the role of IV&V and how IV&V will work with the software safety program and personnel.	NASA-STD-8719.13, 5.2.6.1					x	
Software Safety Requirements	**R.2870**	Describe how safety-critical requirements are generated, implemented, tracked, and verified.	NASA-STD-8719.13, 5.2.6.2					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.2880**	Define the procedures for resolving software safety concerns and recommendations.	NASA-STD-8719.13, 5.2.6.3					x	
Software Safety Requirements	**R.2890**	Describe how software safety and project schedules are synchronized.	NASA-STD-8719.13, 5.2.6.4					x	
Software Safety Requirements	**R.2900**	Specify the number and schedule of software safety assurance audits.	NASA-STD-8719.13, 5.2.6.5					x	
Software Safety Requirements	**R.2910**	Document the conditions requiring software safety engineers to review a situation and proposed solutions or upgrades.	NASA-STD-8719.13, 5.2.6.6					x	
Software Safety Requirements	**R.2920**	Define who monitors system during operation, and what procedures are followed when they feel safety may be threatened.	NASA-STD-8719.13, 5.2.6.7					x	
Software Safety Requirements	**R.2930**	Describe the training requirements for all project software safety roles.	NASA-STD-8719.13, 5.3.1					x	
Software Safety Requirements	**R.2940**	Describe how software safety is integrated with the chosen software life cycle.	NASA-STD-8719.13, 5.5.1					x	
Software Safety Requirements	**R.2950**	List the documents (and associated content) that are part of the software safety program in the Software Safety Plan.	NASA-STD-8719.13, 5.6.1					x	
Software Safety Requirements	**R.2960**	Ensure operational documents describe all safety related commands, data, input sequences, and options.	NASA-STD-8719.13, 7.3					x	
Software Safety Requirements	**R.2970**	Ensure operational documents include error message descriptions and corrective actions.	NASA-STD-8719.13, 7.3.1					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.2980**	Address safety-critical software in all appropriate project documents.	NASA-STD-8719.13, 5.6.3					x	

E.8 SOW CONTENT REQUIREMENTS

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Contract Requirements	**R.2990**	Source code access	NPR 7150.2, SWE-042	x	x				
Software Contract Requirements	**R.3000**	SW measurement data	NPR 7150.2, SWE-044	x	x	x			
Software Contract Requirements	**R.3010**	Insight into test	NPR 7150.2, SWE-039	x	x	x			
Software Contract Requirements	**R.3020**	Electronic access	NPR 7150.2, SWE-040	x	x				
Software Contract Requirements	**R.3030**	Open source	NPR 7150.2, SWE-041	x	x				
Software Contract Requirements	**R.3040**	Track change request	NPR 7150.2, SWE-043	x	x	x			
Software Contract Requirements	**R.3050**	Joint audits	NPR 7150.2, SWE-045	x	x	x			
Software Contract Requirements	**R.3060**	SW schedule	NPR 7150.2, SWE-046	x	x	x			
Software Contract Requirements	**R.3070**	Traceability data	NPR 7150.2, SWE-047	x	x				
Software Contract Requirements	**R.3080**	Solicitation	NPR 7150.2, SWE-048	x	x	x			
Software Safety Requirements	**R.3090**	Contract/MOA/MOU requires safety-critical software be developed according to this Standard.	NASA-STD-8719.13, 5.13.1					x	
Software Safety Requirements	**R.3100**	Software safety deliverables are included in the contract/MOA/MOU.	NASA-STD-8719.13, 5.13.1.1					x	

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Safety Requirements	**R.3110**	Customer surveillance for software safety is included in the contract/MOA/MOU.	NASA-STD-8719.13, 5.13.1.2					x	
Software Safety Requirements	**R.3120**	The contract/MOA/MOU defines how the contractor and customer report and resolve software safety problems.	NASA-STD-8719.13, 5.13.1.3					x	
Software Safety Requirements	**R.3130**	The contract/MOA/MOU defines that customer agreement is required for any changes to baselined safety-critical software elements.	NASA-STD-8719.13, 5.13.1.4					x	

E.9 METRICS REQUIREMENTS

Section	ID	Descriptor	Source	S/W Classification					Comments
				I	II	III	IV	S	
Software Measurement	**R.3150**	SW measurement areas	NPR 7150.2, SWE-091	x	x	x			
Software Measurement	**R.3160**	Collection and storage	NPR 7150.2, SWE-092	x	x	x			
Software Measurement	**R.3170**	Analyze data	NPR 7150.2, SWE-093	x	x				
Software Measurement	**R.3180**	Report analysis	NPR 7150.2, SWE-094	x	x				
Software Measurement	**R.3220**	Objectives	NPR 7150.2, SWE-090	x	x				
Software Report Requirements	**R.3230**	SW Metrics Report	NPR 7150.2, SWE-117	x	x				

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

Status Change	Document Revision	Effective Date	Page	Description of Change
Baseline				
Admin Change	Baseline-1	08/17/10	All	• Formatting changed to comply with Agency standards. The content did not change.