



Langley Research Center

LPR 7150.2

Effective Date: May 14, 2013

Expiration Date: March 31, 2018

## **LaRC Software Engineering Requirements**

**National Aeronautics and Space Administration**

## TABLE OF CONTENTS

P.1 PURPOSE .....	3
P.2 APPLICABILITY.....	3
P.3 AUTHORITY .....	4
P.4 APPLICABLE DOCUMENTS AND FORMS.....	5
P.5 MEASUREMENT/VERIFICATION .....	5
P.6 CANCELLATION .....	6
P.7 TRAINING.....	6
1. INITIATE SOFTWARE ACTIVITY.....	7
1.1 GENERAL REQUIREMENTS.....	7
1.2 SOFTWARE CLASSIFICATION AND SAFETY CRITICALITY .....	7
1.3 FOLLOW THE APPLICABLE SUPPORTING SOFTWARE LMS CPS .....	8
1.4 MULTI-PARTY SOFTWARE ACTIVITIES .....	9
2. TAILORING AND WAIVERS.....	9
2.1 DESIGNATION OF TECHNICAL AUTHORITY BY REQUIREMENT .....	9
2.2 NASA HQ OCE DIRECTION ON TECHNICAL AUTHORITY .....	10
2.3 APPROVAL FOR TAILORING OF SUPPORTING LMS CPS.....	10
2.4 APPROVAL FOR GENERAL EXCLUSION WAIVERS .....	11
3. CENTER LEVEL REQUIREMENTS.....	11
3.1 LARC SENIOR MANAGEMENT .....	11
3.2 LARC SOFTWARE ENGINEERING PROCESS GROUP (SEPG) .....	12
APPENDIX A. DEFINITIONS.....	13
APPENDIX B. ACRONYMS.....	18
APPENDIX C. REFERENCES.....	20
APPENDIX D. NASA-WIDE SOFTWARE CLASSIFICATION DEFINITIONS.....	21
APPENDIX E. COMPLIANCE WITH LAWS, POLICIES, REQUIREMENTS .....	26
APPENDIX F. TECHNICAL AUTHORITY MAPPING MATRIX.....	27

## P.1 PURPOSE

- a. This Langley Procedural Requirements (LPR) (and its supporting Langley Management System Center Procedures (LMS CPs)) defines, clarifies, flows down, and assigns responsibility for the requirements stated in NPR 7150.2A, NASA Software Engineering Requirements, as adapted for Langley Research Center (LaRC) software activities. [SWE-140] [SWE-005] [NPR 7150.2A:6.3.5]

Note: References of the form “[SWE-XXX]” refer to unique software engineering (SWE) requirements numbers in NPR 7150.2; they are included in this LPR and supporting LMS CPs to show traceability to NPR 7150.2 (revision A) and denote Technical Authority delegation (see Section 2 for more details). References of the form “[xxxxxxx]” identify the source from which the text was flowed down.

- b. This LPR provides a single entry point into the LMS system for engineering-related software acquisition, development, maintenance, retirement, operations, management, and assurance activities at LaRC. The blue bolded boxes in Figure 1 illustrate the relationship between this parent LPR document and its supporting Center Procedures (i.e., LMS-CP-7150.3, Class A, B and All Safety-Critical Software; LMS-CP-7150.4, Class C Software; LMS-CP-7150.5, Class D Software; and LMS-CP-7150.6, Class E Software). This LPR specifies requirements applicable to multiple software classes; explains how to determine which supporting LMS CP to follow; and provides the definitions, acronyms, and references used by all the supporting LMS CPs. Each supporting LMS CP specifies requirements related to its respective software class and safety-critical designation. LMS-CP-4754, Software Assurance (SA) for Development and Acquisition, specifies software assurance and safety requirements related to all the software classes.

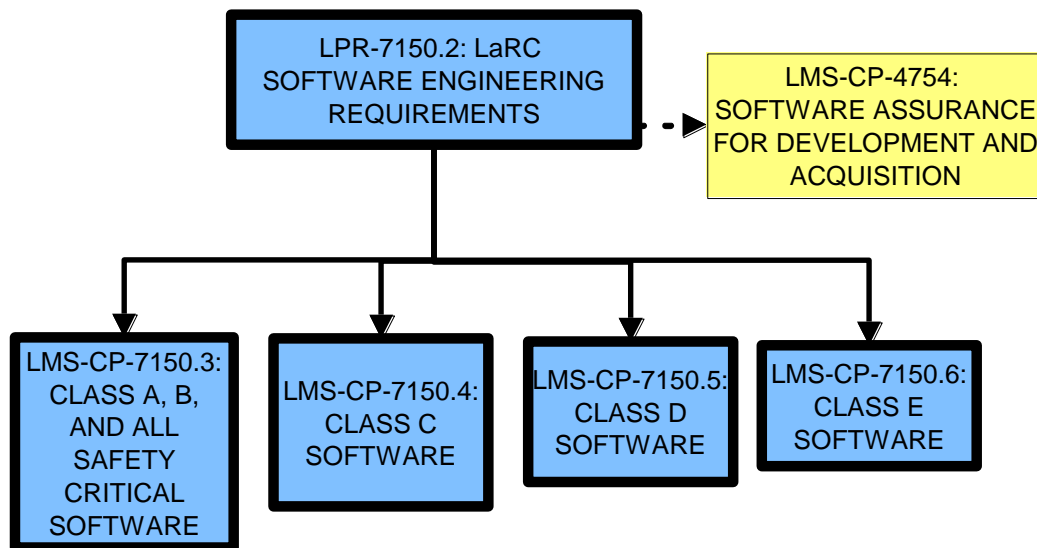


Figure 1. Blue boxes illustrate the relationship between LPR 7150.2 and its supporting LMS CPs.

## P.2 APPLICABILITY

- a. This LPR is applicable to all Class A through E software as defined in Appendix D, NASA-wide Software Classification Definitions, and includes the following types of Class A through E software:
- (1) Software development, maintenance, operations, retirement, management, acquisition, assurance activities and services that are performed, created, or acquired by or for LaRC (hereafter referred to as projects); [SWE-001] [NPR 7150.2A:P.1] this includes new software development, existing software (e.g., commercial-off-the-shelf (COTS) software), modifications to existing software, and the identified software products and associated data, and [NPR 7150.2A:A.30] [NPR 7150.2A:P.1]
  - (2) Software code developed for programmable logic controllers (PLC) and programmable microcode used for microprocessor control, and

- (3) Any software **activities** started after September 27, 2004. [SWE-001]
- b. This LPR applies to the personnel, programs, projects, and tasks at LaRC, including contractors to the extent specified in their respective contracts or agreements. ("Contractors," for purposes of this LPR, include contractors, grantees, Cooperative Agreement recipients, Space Act Agreement partners, or other agreement parties.) [NPR 7150.2A:P.2.2]
- c. This LPR and supporting LMS CPs are made applicable to contractors through contractual clauses, specifications, or statements of work in conformance with the NASA Federal Acquisition Regulation (FAR) Supplement or through grants or other party agreements. [NPR 7150.2A:P.2.2] [NPR 7150.2A:P.2.4] [NPR 7150.2A:6.3.5]
- d. Exclusions:

- (1) The gate level structure (hardware aspects) of field programmable gate arrays (FPGA), complex programmable logic device (CPLD), system-on-chip (SoC), and application-specific integrated circuits (ASIC) are not subject to this LPR. Due to differing engineering processes, these devices shall be subject to a tailored flight hardware development plan using guidance from NASA-HDBK 8739.23, NASA Complex Electronics Handbook for Assurance Professionals, until further NPR/LPR direction is provided. Processor cores can be incorporated into complex electronic devices (FPGA, ASIC, SoC). Though the processor core falls under this exclusion, machine or byte code that runs on the processor core is software, and the LPR applies to the code. This remains true in the case where the code is stored on-chip.

Note:

- C code running on the soft core in an FPGA is software (it is not part of the FPGA because you can burn a core into an FPGA);
- You can also set aside a portion of the FPGA to act as an on-chip memory and load software onto that memory;
- All of the above are software and subject to this LPR. The development process for the processor core is distinctly different from the development process for C Code, and the C Code is in scope of the LPR. The complex electronic device is restricted to the gate level structure of the device, not software loaded onto a core that is burned onto the device.

- (2) Information technology systems as defined in NPR 7150.2, Classes F, G, and H. [NPR 7150.2A:2.Appendix C]
- (3) Stand-alone desktop applications (e.g., word processing programs, project scheduling software, presentation programs). [NPR 7150.2A:2.3]
- (4) This LPR does not apply if solely acquiring a standalone COTS, government-off-the-shelf (GOTS), reused, or open source software product to satisfy requirements; no software development or modifications will be performed in the software activity/project; no data produced by the software will be incorporated or used with or within a NASA system (i.e., acquiring a standalone, completed software product that will not be modified and will not be included within a NASA application; and whose resulting data is not included in nor used to engineer, verify and validate, support operations of, or maintain a NASA system). [NPR 7150.2A:2.3 modified]

Note: Follow LMS-CP-4501, Procurement Process Overview to procure COTS products.

- (5) This LPR does not apply to non-safety critical standalone software or related output data that:
- (a) Has no anticipated delivery, and;
  - (b) Is not the subject of a publication or delivered/published analyses, and
  - (c) Is not included in a system that is being delivered, and
  - (d) Will not be used to make decisions on Class A, B, C, or D systems.
- (6) If a project predates the LPR release, from the LPR release date forward, the project will follow the LPR requirements for the current and all future project activities (i.e., if the project started before the LPR was approved and it does not make sense to go back and fulfill requirements from completed phases, then the project complies with the LMS requirements for the present and remaining project phases). This will be documented in the project's Compliance Matrix.

### P.3 AUTHORITY

Note: NPDs and NPRs are found in the NASA Online Directives Information System (NODIS) at: <http://nodis3.gsfc.nasa.gov/>.

- a. NPD 1280.1, NASA Integrated Management System Policy.
- b. NPR 7150.2, NASA Software Engineering Requirements.

#### **P.4 APPLICABLE DOCUMENTS AND FORMS**

Note: NPDs and NPRs are found in the NASA Online Directives Information System (NODIS) at: <http://nodis3.gsfc.nasa.gov/>. Standards are found on the NASA Technical Standards Program Web site at: <https://standards.nasa.gov/>. NASA forms are found at: <http://server-mpo.arc.nasa.gov/services/NEFS/>. The latest versions of the following documents are to be applicable. In the event of conflict among the top-level directives and one or more lower-level directives or procedures, the information provided in the top-level directive takes precedence. In the event of a conflict among documents at the same level, consult the Designated Technical Authority for resolution.

- a. Electronic and Information Technology Accessibility Standards, 36 CFR Part 1194.
- b. NPD 2091.1, Inventions Made by Government Employees.
- c. NPD 2810.1, NASA Information Security Policy.
- d. NPD 9250.1, Capital Asset Identification and Treatment.
- e. NPR 2190.1, NASA Export Control Program.
- f. NPR 2210.1, Release of NASA Software.
- g. NPR 7120.5, NASA Space Flight Program and Project Management Requirements.
- h. NPR 7120.7, NASA Information Technology and Institutional Infrastructure Program and Project Management Requirements.
- i. NPR 7120.8, NASA Research and Technology Program and Project Management Requirements.
- j. NPR 7123.1, NASA Systems Engineering Processes and Requirements.
- k. NPR 7150.2, NASA Software Engineering Requirements.
- l. NPR 8000.4, Agency Risk Management Procedural Requirements.
- m. NPR 8709.20A, Management of Safety and Mission Assurance Technical Authority (SMA TA) Requirements.
- n. NPR 8715.3, NASA General Safety Program Requirements.
- o. NPR 8735.2, Management of Government Quality Assurance Functions for NASA Contracts.
- p. LAPD 1150.2, Councils, Boards, Panels, Committees, Teams, and Groups.
- q. LAPD 2810.1, Security of Information Technology.
- r. LPR 1620.1, Information Security Program Management Procedures and Guidelines.
- s. LPR 7120.4, Langley Research Center Technical Authority Implementation Plan.
- t. NASA-STD-8709.20, Management of Safety and Mission Assurance Technical Authority (SMA TA) Requirements.
- u. NASA-STD-8719.13, Software Safety Standard.
- v. NASA-STD-8739.8, Software Assurance Standard.
- w. NF 1739, Alternative Future Use Questionnaire - NASA Projects.
- x. LMS-CP-1723, External Release of NASA Software.
- y. LMS-CP-7150.3, Class A, B, and All Safety-Critical Software.
- z. LMS-CP-7150.4, Class C Software.
- aa. LMS-CP-7150.5, Class D Software.
- bb. LMS-CP-7150.6, Class E Software.
- cc. LMS-CP-7151, Obtaining Waivers for Langley Management System (LMS) Requirements.
- dd. LMS-CP-4501, Procurement Process Overview.
- ee. LMS-CP-4754, Software Assurance (SA) for Development and Acquisition.
- ff. LMS-CP-5523, Statement of Work (SOW) Review Procedure.
- gg. LMS-CP-5524, Product Requirements Review Procedure (for Low Control).
- hh. LMS-CP-5526, Product Requirements Development and Management Procedure.
- ii. NASA-SP-2007-6105, NASA System Engineering Handbook, <https://nen.nasa.gov/syseng/NASA-Systems-Engineering-Handbook.pdf>.
- jj. Peer Review Toolkit, <https://sites-e.larc.nasa.gov/sweng/supporting-products/>.
- kk. Additional Supporting Information and Documents for Software Engineering and LMS Software Procedures: <https://sites-e.larc.nasa.gov/sweng/supporting-products/>.

#### **P.5 MEASUREMENT/VERIFICATION**

- a. Verification will be accomplished by inspection as part of the LaRC Internal Audit process.

- b. The NASA Headquarters (HQ) Office of the Chief Engineer (OCE) authorizes appraisals against this LPR and supporting LMS CPs to check compliance. [SWE-129]
- c. Compliance with this LPR and supporting LMS CPs is documented in the software Compliance Matrix maintained by each software project (see supporting LMS CPs for details).
- d. The LaRC Software Engineering Process Group (SEPG), as delegated by the LaRC Center Director, keeps a copy of all projects' software Compliance Matrices containing approved tailoring and waivers. [SWE-128]
- e. The LaRC Representative to the (Agency) Software Working Group retains the list of LaRC software projects and updates the list every year in the NASA Software Inventory. [SWE-006] (See [https://sites-e.larc.nasa.gov/sweng/home\\_pg/](https://sites-e.larc.nasa.gov/sweng/home_pg/) for the name of the current representative.)
- f. On an annual basis, the SEPG analyzes data entered into the Langley Software Metrics Collection database and reports results to the LaRC Chief Engineers Board. [SWE-094]

#### **P.6 CANCELLATION**

- a. LMS-CP-5528, Software Planning, Development, Acquisition, Maintenance, and Operations.
- b. LMS-CP-5529, Software Configuration Management Planning for Low-, High-, and Critical-Control Software.
- c. LMS-CP-5532, Software Acquisition Planning.

#### **P.7 TRAINING**

- a. For answers to questions or to request training on this LPR and its supporting LMS CPs, send an email to: [larc-dl-support-sepg-help](mailto:larc-dl-support-sepg-help) or call the LMS Software procedure help desk phone number provided at [https://sites-e.larc.nasa.gov/sweng/home\\_pg/](https://sites-e.larc.nasa.gov/sweng/home_pg/).
- b. The HQ OCE maintains an Agency-wide process asset library of software engineering best practices (e.g., example plans, software estimating tool, coding standards, requirements traceability matrix). Projects can review the contents to identify applicable practices that may add value to their software activities. See: <https://nen.nasa.gov/web/software/nasa-software-process-asset-library-pal>. [SWE-099]

*Original signed on file*

Stephen G. Jurczyk  
Deputy Director [SWE-122]

#### **DISTRIBUTION**

Approved for public release via the Langley Management System; distribution is unlimited.

## 1. INITIATE SOFTWARE ACTIVITY

### 1.1 General Requirements

- 1.1.1 Projects shall fully comply with the requirements in this LPR and supporting LMS CPs. [SWE-139]  
 Note: Additionally, projects identify and comply with the applicable laws, policies, and requirements in Appendix E.
- 1.1.2 The NASA line manager responsible for the software tasks shall assign a software manager for the project.
- 1.1.3 In each of this LPR's supporting LMS CPs, a software Compliance Matrix is produced against the supporting CP.
- At the LaRC 60-Day Review (if held), the project shall report when the software Compliance Matrix will be completed and reviewed.
  - If a 60-Day Review is not held for a project, this information shall be reported at the first periodic review (reference LPR 7130, Project and Task Review Procedural Requirements, Section 3.j).
  - To assure compliance with this LPR and its supporting LMS CPs, the Mission Assurance Branch shall perform spot check audits between the software Compliance Matrix and non-safety-critical projects with non-space flight Class C software and Class D software. [SWE-032] [SWE-022]
  - For all software that is Class A, Class B, space flight software and the ground software that controls or monitors space flight Class C, and safety-critical software, the Mission Assurance Branch shall assure compliance with requirements and standards as specified in LMS-CP-4754, Software Assurance (SA) for Development and Acquisition. [SWE-022] [SWE-032]

### 1.2 Software Classification and Safety Criticality

- 1.2.1 The software manager shall classify each system and subsystem containing software in accordance with the software classification definitions for Classes A, B, C, D, and E specified in Appendix D; the classification applies to acquired, developed, reused, or maintained software. [SWE-020] [SWE-001]  
 Note:
- The system or subsystem in which the software operates is classified (as Class A through E), not the software itself (see definition of system and subsystem in Appendix A). Classifications are defined by intended function. Software is only a logical description of a function and requires the remainder of the system (e.g., hardware, other software, and data) to realize the function. [NPR 7150.2A: NODIS Comment #224]
  - A given system or subsystem is uniquely defined within a single class. [NPR 7150.2A:Appendix E Modified]
  - Some projects may contain multiple systems and subsystems having different software classes. [NPR 7150.2A:P.2.1]
  - The number of applicable requirements and their associated rigor are scaled back for lower software classes and software designated as non-safety-critical. [NPR 7150.2A: Appendix E Modified]
  - When a COTS, GOTS, modified-off-the-shelf (MOTS), reused, or open source software component (e.g., math libraries, databases) is included within a NASA application, the software component is assessed and classified as part of that system or subsystem in which the application resides. [NPR 7150.2A: 2.3 Modified]
  - Software that supports development activities (e.g., verification software, simulations) can be classified separately from the system under development; however, such support software may be developed in conjunction with the system and not as a separate project. Therefore, projects can use the same processes and work products established for the development of the system (e.g., Class A flight software) to satisfy the LMS requirements for the development of the support software (e.g., Class C hardware/software integration lab). [NPR 7150.2A:2.2.8.1 Modified]
- 1.2.2 If more than one software class seems to apply to a given system or subsystem, then the software manager shall assign the higher of the classes to the system/subsystem (classes are in descending order from A to E). [NPR 7150.2A:Appendix E Modified]  
 Note: Any discrepancies/uncertainties in classifying software within Classes A through E are to be resolved using the definitions in Appendix D and the following five underlying factors. The software class definitions are based on 1) usage of the software with or within a NASA system, 2) criticality of the system to NASA's major programs and projects, 3) extent to which humans depend upon the

system, 4) developmental and operational complexity, and 5) extent of the Agency's investment.

[NPR 7150.2A: Appendix E Modified]

**1.2.3 Following LMS-CP-4754, Software Assurance (SA) for Development and Acquisition:**

- a. The software manager shall notify the Head, Mission Assurance Branch (MAB), of the software activity and the associated software classification(s).
- b. MAB shall perform an independent classification assessment of the systems and subsystems that contain software. [SWE-132]
- c. The software manager, in conjunction with MAB, shall determine the software safety criticality in accordance with NASA-STD-8739.8, Software Assurance Standard. [SWE-133]

Note:

- Software safety criticality is initially determined in the formulation phase. As the software is developed or changed and the software items, models, and simulations are identified, the safety-critical software determination can be reassessed and applied at lower levels. The software safety assessment is performed for each software acquisition, development, and maintenance activity, and for changes to legacy/heritage systems. [NPR 7150.2A:2.2.8.3]
- As some projects may have multiple software tasks, each may need to be assessed separately.

**d. MAB shall:**

- (1) Record the results of the software safety assessment and the independent assessment of software class(es) in the Software Assurance Classification Assessment Report, and [NASA-STD-8739.8:5.1.2.1] [NASA-STD-8739.8:Appendix A] SWE-022]
  - (2) Configuration manage and maintain the report as a quality record. [NASA-STD-8739.8:6.6.1] [NASA-STD-8739.8:5.2.1.6] [SWE-022]
- e. MAB shall ensure that the Software Assurance Classification Assessment Report is made available to the Director, Safety and Mission Assurance (SMA) Office at LaRC; project management; and/or LaRC Center Director upon request. [NASA-STD-8739.8:5.2.1.6] [NASA-STD-8739.8:6.6.1] [SWE-022]

**1.2.4 MAB and the software manager shall compare the independent classification against the project classification and reach agreement.** [NASA-STD-8739.8:5.1.2.1] [NASA-STD-8739.8:5.1.2.4] [NPR 7150.2A:2.2.8.1 Modified] [SWE-022]

**1.2.5 Disagreement over software classification or safety criticality determination shall be resolved using the dissenting opinion section of LPR 7120.4, Langley Research Center Technical Authority Implementation Plan.** [LPR 7120.4:10] [NASA-STD-8739.8:5.1.2.4] [SWE-022]

**1.2.5.1 Engineering and SMA provide dual Technical Authority chains for resolving classification issues.**

[NPR 7150.2A:Appendix E] Disagreements are elevated via both the Engineering Technical Authority (the software manager's Directorate Head) and SMA Technical Authority (Director, SMA Office at LaRC). [NPR 7150.2A:2.2.8.1 Modified] The NASA HQ Chief Engineer (CE) is the ultimate Technical Authority for software classification disputes concerning definitions in this LPR. Jointly, NASA HQ CE and NASA HQ Chief, SMA are ultimately the Technical Authorities for safety criticality disputes. [NPR 7150.2A:Appendix E]

**1.2.6 If a system or subsystem evolves to a higher software classification, or changes from non-safety-critical to safety-critical, then the software manager shall:**

- a. Return to this LPR 7150.2 to repeat completion of section 1.2 and 1.3.
- b. Fill out the Class-specific Compliance Matrix for the new software Class. [SWE-125]
- c. Update plans to fulfill the added requirements specified for the higher software Class and/or safety criticality. [SWE-021]
- d. Gain approval of the updated Software Management Plan and the new Compliance Matrix by following the approval steps in the applicable supporting LMS CP specified in Section 1.3 and complete any other applicable section of this LPR. [SWE-026]
- e. Provide the updated software inventory data for the project (e.g., software class, safety criticality) to LaRC's current representative to the Agency Software Working Group. (See [https://sites-e.larc.nasa.gov/sweng/home\\_pg/](https://sites-e.larc.nasa.gov/sweng/home_pg/) for name of current representative.) [SWE-006] [SWE-131]

**1.3 Follow the Applicable Supporting Software LMS CPs**

**1.3.1 Based on the software classification and safety criticality of the software (determined in Section 1.2), the software manager shall follow the LMS CP(s) that apply:**

- a. LMS-CP-7150.3: Class A, B and All Safety-Critical Software – applies for all Class A and Class B software and all software that is safety-critical.

- b. LMS-CP-7150.4: Class C Software – applies for all Class C software that is not safety-critical.
- c. LMS-CP-7150.5: Class D Software – applies for all Class D software that is not safety-critical.
- d. LMS-CP-7150.6: Class E Software – applies for all Class E software that is not safety-critical.

#### 1.4 Multi-Party Software Activities

This section pertains to support, partnership, and inter-Center agreements. These are frequently referred to as “Memorandums of Agreements.” This section does not pertain to contractual agreements.

- 1.4.1 When software development, maintenance, operations, management, acquisition, or assurance activities are being conducted in collaboration with non-NASA parties, the following shall be documented in a shared agreement with responsibilities for each identified: [SWE-125]
  - a. The technical requirements to be performed or a reference to them; [SWE-049]
  - b. The LMS CP requirements to be performed by each party as defined in the approved software Compliance Matrix; [SWE-125]
  - c. If the software is class A, B, or C or the software is safety-critical, the assurance requirements defined in the NASA-STD-8739.8: Software Assurance Standard to be performed by each party (NASA-STD-8739.8, Appendix C contains a Requirements Compliance Matrix that may be used to document the responsible parties for each requirement); and [SWE-022]
  - d. If the software is safety-critical, the safety requirements defined in NASA-STD-8719.13: Software Safety Standard to be performed by each party (NASA-STD-8719.13, Appendix B-2 contains a Requirements Compliance Matrix that may be used to document the responsible parties for each requirement). [SWE-023]

##### Note:

- If LaRC is performing the software activity or service at the request of another Center, the software manager assures the requirements in Section 1.4.1 to be fulfilled by LaRC are documented in the support, partnership, or inter-Center agreement before it is signed.
- When software work is being conducted by another Center or party at LaRC's request, the software manager documents in the support, partnership, or inter-Center agreement, the requirements in Section 1.4.1. [SWE-125]
- If the request from another Center or party is not for software products or services but LaRC chooses to conduct software activities in fulfilling the request, then those software activities are treated as a LaRC software project and the software manager assures: (1) that the technical requirements to be fulfilled by LaRC are documented or referenced in the support, partnership, or inter-Center agreement, and (2) that the LaRC software project complies with this LPR and its supporting LMS CPs.

## 2. TAILORING AND WAIVERS

In this section, “LPR” refers to this LPR and its supporting LMS CPs.

### 2.1 Designation of Technical Authority by Requirement

- 2.1.1 For LPR requirements appended with a reference of the form “[SWE-XXX]”:
  - a. The reference refers to the parent NPR 7150.2 software engineering (SWE) requirement number.
  - b. Appendix F of this LPR lists the “SWE” requirement numbers from NPR 7150.2 (revision A) and shows the required Designated Technical Authority<sup>1</sup> for approving any tailored or waived requirements. [NPR 7150.2A:Appendix D] [LPR 7120.4:7.3] [SWE-122]
  - c. Additional approvals required: [NPR 7150.2A:Appendix D, Note 3]
    - (1) Tailoring and waivers that involve safety-critical software shall have co-approval from the Director, SMA Office at LaRC<sup>1</sup>.
    - (2) Tailoring and waivers that involve software with health and medical implications shall have co-approval from the Occupational Health Contracting Officer Technical Representative in the Morale, Welfare, and Recreation Branch within the Office of Human Capital Management.

<sup>1</sup> As per LPR 7120.4, the LaRC Center Director has delegated Technical Authority to the Directorate Heads for the engineering directorates and to Director, SMA Office at LaRC for safety and mission assurance. [LPR 7120.4:7.3 & 7.6]

2.1.2 For tailoring/waivers against LPR requirements without a reference of the form '[SWE-XXX]', the Designated Technical Authority for approval shall be the software manager's Directorate Head.

[LPR 7120.4:7.3]

2.1.3 For tailoring/waivers against LPR requirements that reference LMS-CP-4754, the Designated Technical Authority for approval shall be the Director, SMA Office<sup>1</sup> at LaRC. [LPR 7120.4:7.6]

Note: For tailoring/waivers against LPR requirements that reference NASA-STD-8739.8, NASA Software Assurance Standard, and NASA-STD-8719.13, Software Safety Standard, refer to NPR 8709.20 Management of Safety and Mission Assurance Technical Authority (SMA TA) Requirements, which specifies allowed tailoring and also the process for adjudication of requests for relief from SMA requirements needing NASA HQ Chief, SMA approval.

## 2.2 NASA HQ OCE Direction on Technical Authority

2.2.1 The software manager's Directorate Head shall consider the following information when assessing tailoring/waiver requests: [SWE-124] [SWE-126]

- a. The NASA software inventory data on the project (go to [https://sites-e.larc.nasa.gov/sweng/home\\_pg/](https://sites-e.larc.nasa.gov/sweng/home_pg/) and contact LaRC's current representative to the Agency Software Working Group to obtain the latest copy of the software inventory).
- b. The classification of systems and subsystems containing software, as defined in Appendix D.
- c. This LPR and its supporting LMS CPs.
- d. Applicable contractor and subcontractor software policies and procedures that meet the intent of this LPR and its supporting LMS CPs.
- e. Potential impacts to NASA missions.
- f. Potential impacts to health, medical concerns, or safety.

Note: The NASA HQ OCE will authorize appraisals to check compliance against this LPR's requirements and review approved tailoring and waivers. [SWE-129]

## 2.3 Approval for Tailoring of Supporting LMS CPs

2.3.1 The software manager shall document tailoring requests in the project's software Compliance Matrix as an appendix to the Software Management Plan; each supporting LMS CP contains a Class-specific Compliance Matrix that specifies required information to be recorded for tailoring requests.

2.3.2 For the purposes of this LPR, tailoring is modification or deletion of anything required by the LMS CPs (including items in the flow diagrams, Step-Action Tables, Appendices, and items in subelements/subparagraphs).

Exception: Items that are marked with an asterisk "\*" may be denoted as "not applicable" (NA) in the Compliance Matrix with no tailoring approval required.

2.3.3 The software manager shall obtain approvals from the following officials for tailoring requests defined in the software Compliance Matrix:

- a. Software manager.
- b. Applicable project personnel (individuals accepting the risk associated with the tailoring).
- c. Software manager's line manager.
- d. Mission Assurance Branch.
- e. Directorate's Software Engineering Process Group (SEPG) representative (concurrence only). [NPR 7150.2A:Note 6.2.1]
- f. Designated Technical Authority (i.e., the software manager's Directorate Head).

2.3.4 The Designated Technical Authority shall take into account the SEPG inputs. [NPR 7150.2A:Note 6.2.1]

2.3.5 For tailoring that requires "NASA HQ Chief Engineer" approval (as designated in the last column of Appendix F), the Directorate Head shall process the request through LMS-CP-7151, Obtaining Waivers for Langley Management System (LMS) Requirements, prior to submitting the request for approval to HQ CE (with a copy to HQ Deputy CE). [NPR 7150.2A:Appendix D]

2.3.6 For tailoring that requires both "NASA HQ Chief Engineer" approval and "NASA HQ Chief, Safety and Mission Assurance" approval (as designated in the last column of Appendix F), the software manager's Directorate Head jointly with the Director, SMA Office at LaRC, shall approve the request and process it through LMS-CP-7151 prior to submitting the request for approval to the HQ CE (with a copy to HQ Deputy CE) and to the HQ Chief, SMA. [NPR 7150.2A:Appendix D] [LPR 7120.4:4] [SWE-001]

2.3.7 After initial approval, if the project desires to modify the approved Compliance Matrix, the software manager shall resubmit the Software Management Plan that contains the Compliance Matrix, with the requested matrix modifications clearly identified, to the Designated Technical Authority.

#### 2.3.7.1 The Designated Technical Authority shall:

- a. Assist the software manager in processing the request for Compliance Matrix modifications within the Software Management Plan using LMS-CP-7151, and
- b. Use the same list of officials involved during the initial matrix approval (see 2.3.3).

#### 2.3.8 The SEPG shall forward an information copy of all software Compliance Matrices containing tailoring and waivers approved at the Center level to the HQ CE (with a copy to the HQ Deputy CE). [HQ/OCE Delegation Letter 10/14/08] [SWE-128]

### 2.4 Approval for General Exclusion Waivers

General exclusion waivers apply to LPR and supporting LMS CP requirements appended with the reference of the form [SWE-XXX]. For assistance contact the LaRC SEPG Chair.

#### 2.4.1 For those cases in which an organization or multiple instances/teams desires (1) a general exclusion from LPR requirement(s) or (2) to generically apply specific alternate requirements that do not meet or exceed this LPR's requirements, the requester shall submit a waiver following LMS-CP-7151 for those exclusions or alternate requirements with appropriate justification to the Directorate Head for concurrence. [SWE-120]

Note: In other words, this type of waiver (which is ultimately approved by the NASA HQ CE) is for generic or blanket relief from an LPR requirement for a Center organization, multiple teams, or multiple projects over an extended time. General or blanket waivers are not to be confused with normal tailoring that addresses relief from an LPR requirement on a single project or in a specific instance as covered in Section 2.3. [NPR 7150.2A:6.1.1Modified]

#### 2.4.2 If approved, the Directorate Head shall submit the general exclusion waiver with appropriate justification for approval to the NASA HQ CE (and a copy to HQ Deputy CE). [SWE-120] [LPR 7120.4.4]

Note: If multiple projects under an organization individually submit tailoring for the same LPR requirement(s), the Directorate Head processes the collection of tailoring items through LMS-CP-7151 and submits the collection as a general exclusion waiver, with appropriate justification, for approval to the HQ CE (and a copy to HQ Deputy CE).

#### 2.4.3 The general exclusion waiver shall contain, at a minimum, the Center and project name, the software class(es) that the request applies to, the applicable LMS document and element or subelement number of the requirement(s) for which the waiver is requested, associated [SWE-XXX] reference, an explanation of the waiver requested, the impacts/risks associated with the waiver, and the justification (why the impacts and risks are acceptable). [HQ/OCE Delegation Letter 10/14/08]

#### 2.4.4 The Directorate Head shall ensure that when general exclusion waivers are approved by the NASA HQ CE:

- a. The requesting organizations or projects/teams shall document the approved alternate requirements in an LMS procedure/work instruction or the project/team documentation controlling the development, acquisition, and/or deployment of the affected software, and [SWE-121]
- b. A copy of the approved general exclusion waiver shall be sent to the SEPG via email at [LaRC-DL-SW-Waiver@mail.nasa.gov](mailto:LaRC-DL-SW-Waiver@mail.nasa.gov). [SWE-128]

## 3. CENTER LEVEL REQUIREMENTS

### 3.1 LaRC Senior Management

#### 3.1.1 The LaRC Center Director, Engineering Directors, and Chief Engineer shall:

- a. Be responsible for ensuring all LaRC software activities are in compliance with this LPR and supporting LMS CPs, and
- b. Provide resources to maintain, staff, and implement a plan to continually advance LaRC in-house software engineering capability and monitor the software engineering capability of NASA's contractors (see 3.2.2 for plan required content). [SWE-003] [SWE-108]

#### 3.1.2 Each Engineering Directorate Head shall appoint a Software Engineering Process Group representative who is a recognized software engineering expert to support the Directorate Head in performing Technical Authority software-related functions. [NPR 7150.2A: Note 6.2.1]

#### 3.1.3 The LaRC Organizational Unit Managers shall, every other year, submit to the LaRC Software Working Group representative the NASA Software Inventory information requested. [SWE-006]

#### 3.1.4 The LaRC Training, Development, and Employee Relations Branch shall provide and fund training to advance software engineering practices and software acquisition. [SWE-100]

### 3.2 LaRC Software Engineering Process Group (SEPG)

Refer to LAPD 1150.2 Councils, Boards, Panels, Committees, Teams, and Groups for the SEPG charter.

- 3.2.1 The SEPG shall maintain, staff, and implement the Center Plan for LaRC Software Process Improvement to continually advance its in-house software engineering capability and monitor the software engineering capability of NASA's contractors. [SWE-003] [SWE-108] Go to [https://sites-e.larc.nasa.gov/sweng/home\\_pg/](https://sites-e.larc.nasa.gov/sweng/home_pg/) to see the latest plan.
- 3.2.2 The Center Plan for LaRC Software Process Improvement shall include: [SWE-108]
  - a. Process improvement goal(s).
  - b. Scope of process improvement.
  - c. All Center organizations responsible for the performance of mission-critical software development, management, and acquisition.
  - d. The Center's tactic for phasing in improvements (e.g., domain phasing and organizational phasing).
  - e. Ownership of Center Software Engineering Improvement Plan.
  - f. The Center's tactic for monitoring Center Software Engineering Improvement Plan progress, including responsibilities.
  - g. Strategies and objectives.
  - h. The Center's tactic for supporting the implementation of all strategies of the NASA Software Engineering Initiative Implementation Plan.
  - i. Schedule.
  - j. The Center's tactic or approach for phasing in new and upgraded NASA HQ requirements.
- 3.2.3 The SEPG shall maintain and implement a Software Training Plan to advance LaRC's in-house software engineering capability and as a reference for its contractors. [SWE-101]
- 3.2.3.1 The LaRC Software Training Plan shall include the following: [SWE-107] [SWE-101]
  - a. Responsibilities.
  - b. Implementation.
  - c. Records and forms.
  - d. Training resources.
  - e. Minimum training requirements for software personnel.
  - f. Training class availabilities.

Note: This plan is typically included within the Center Plan for LaRC Software Process Improvement.

## Appendix A. DEFINITIONS

This Appendix of LPR-7150.2 provides the definitions for this LPR and all the LMS-CP-7150.x supporting procedures.

- A.1 **Acceptance testing:** (1) Formal testing conducted to determine whether a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system. (2) Formal testing conducted to enable a user, customer, or other authorized entity to determine whether to accept a system or component. [IEEE STD 610.12-1990]
- A.2 **Accuracy:** The difference between a parameter or variable (or a set of parameters or variables) within a model, simulation, or experiment and the true value or the assumed true value. (Definition from source document: NASA-STD-7009, Standard for Models and Simulations.) [NPR 7150.2A:A.1]
- A.3 **Accredit:** The official acceptance of a software development tool, model, or simulation, (including associated data) to use for a specific purpose. [NPR 7150.2A:A.2]
- A.4 **Airborne Vehicle (Aircraft):** Any structure, machine, or contrivance, especially a vehicle, designed to be supported by the air, being borne up either by the dynamic action of the air upon the surfaces of the structure or object, or by its own buoyancy; such structures, machines, or vehicles collectively, as, fifty aircraft. Aircraft, in its broadest meaning, includes fixed-wing airplanes, helicopters, gliders, airships, free and captive balloons, ornithopters, flying model aircraft, kites, etc., but since the term carries a strong vehicular suggestion, it is more often applied, or recognized to apply, only to such of these craft as are designed to support or convey a burden in or through the air. [Aerospace Science and Technology Dictionary]
- A.5 **Analysis:** (1) The post-processing or interpretation of the individual values, arrays, files of data, or execution information. [NPR 7150.2A:A.3] (2) The processing of accumulated data obtained from other verification methods. Examples are reduction, interpolation, or extrapolation of test results. [J-STD-016:F.2.2(4)]
- A.6 **Configuration item:** An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process. [IEEE 610.12-1990]
- A.7 **Computer software configuration item (CSCI):** An aggregation of software that is designated for configuration management and treated as a single entity in the configuration management process. [IEEE 610.12-1990]
- A.8 **Contracted Software:** Software created for a project by a contractor or subcontractor. [NPR 7150.2A:A.4]
- A.9 **Contractual agreement:** Types include contracts, task orders, task agreements, etc. Contractual agreements contain statements of work (SOW), solicitation, requests for proposals (RFP) or similar document(s) which define the work to be performed by the contractor. [LMS]
- A.10 **Data:** Information for computer processing (e.g., numbers, text, images, and sounds in a form that is suitable for storage in or processing by a computer). [NPR 7150.2A:A.5]
- A.11 **Demonstration:** The operation of the software item, or a part of the software item, that relies on observable functional operation not requiring the use of instrumentation, special test equipment, or subsequent analysis. [J-STD-016:F.2.2(4)]
- A.12 **Delivery:** Release of a system or component to its customer or intended user. [IEEE-610.12]
- A.13 **Documentation tree:** A diagram that depicts all of the documents for a given system and shows their relationships to one another. [IEEE 610.12-1990]
- A.14 **Embedded Software:** Software that is part of a larger system and performs some of the requirements of that system (Definition from source document: ISO/IEC 24765:2009 Systems and Software Engineering Vocabulary.). [NPR 7150.2A:A.7]
- A.15 **End-to-end testing:** The objective of end-to-end testing is to demonstrate interface compatibility and desired total functionality among different elements of a system, between systems, and systems as a whole. Interfaces include software/software, hardware/software, and system/system data exchanges. In addition, end-to-end testing includes complete operational scenarios across system components to verify that performance requirements are met. End-to-end testing verifies that the data flows throughout the multi-system environment are correct, the system provides the required functionality, and that the outputs at the eventual end points correspond to expected results. [Based on draft NPR2820]
- A.16 **Establish and Maintain:** The responsible project, organization, or individual must formulate, document, use/deploy, and keep current the object (usually a document, requirement, process, or policy). [NPR 7150.2A:A.8]

- A.17 **Firmware:** The combination of a hardware device and computer instructions and data that reside as read-only software on that device. [IEEE 610.12-1990]
- A.18 **Function:** A group of related requirements. The word “function” may be replaced with “capability,” “subject,” “object,” or other term useful for presenting the requirements. [J-STD-016:F.2.4(3.2)]
- A.19 **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. [NPR 7150.2A:A.9]
- A.20 **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 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.) [NPR 7150.2A:A.10]
- A.21 **Insight:** Surveillance mode requiring the monitoring of customer-identified metrics and contracted milestones. Insight is a continuum that can range from low intensity, such as reviewing quarterly reports, to high intensity, such as performing surveys and reviews. (Definitions from source document: NPR 8735.2, Management of Government Safety and Mission Quality Assurance Surveillance Functions for NASA Contracts.) [NPR 7150.2A:A.11]
- A.22 **Independent Verification and Validation (IV&V):** Verification and validation performed by an organization that is technically, managerially, and financially independent of the development organization (ISO/IEC 24765:2008 systems and software engineering vocabulary). [NPR 7150.2A:A.12]
- A.23 **Legacy/Heritage:** 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. See software reuse. [NPR 7150.2A:A.13]
- A.24 **Life cycle model:** A partitioning of the life of a product or project into phases. [CMMI V1.2]
- A.25 **Major Engineering/Research Facility:** Used in this document to show research, development, test, or simulation facilities representing a significant NASA investment which contains software that supports programs and projects managed under NPR 7120.5, NPR 7120.7, or NPR 7120.8. Examples include: high-fidelity, motion-base flight simulator facilities (e.g., Vertical Motion Simulator at Ames), wind tunnels (e.g., National Transonic Facility at LaRC), vacuum chambers (e.g., Space Power Facility at GRC/Plum Brook), air traffic control facilities (e.g., North Texas Research Station), and engine test stands (e.g., J-2X test stand at Stennis Space Center). [NPR 7150.2A:A.14]
- A.26 **Maintenance:** The process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment. [IEEE 610.12-1990]
- A.27 **Mathematical Model:** The mathematical equations, boundary values, initial conditions, and modeling data needed to describe the conceptual model (ASME V&V 10). (Definition from source document: NASA-STD-7009, Standard for Models and Simulations.) [NPR 7150.2A:A.15]
- A.28 **Mission Critical:** Item or function that must retain its operational capability to assure no mission failure (i.e., for mission success). (Definition from source document: NPR 8715.3, NASA General Safety Program Requirements.) [NPR 7150.2A:A.16]
- A.29 **Model:** A description or representation of a system, entity, phenomena, or process. (Definition from source document: NASA-STD-7009, Standard for Models and Simulations.) Only for the purpose of this document, the term “model” refers only to those models which are implemented in software. [NPR 7150.2A:A.17]
- A.30 **Modified Off-the-Shelf (MOTS) Software:** When COTS, legacy/heritage software is reused, or heritage software is changed, the product is considered “modified.” The changes can include all or part of the software products and may involve additions, deletions, and specific alterations. An argument can be made that any alterations to the code and/or design of an off-the-shelf software component constitutes “modification,” but 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. See “off-the-shelf.” [NPR 7150.2A:A.18]
- A.31 **Off-the-Shelf Software:** Software not developed in-house or by a contractor for the specific project now under way. The software is generally developed for a different purpose than the current project. [NPR 7150.2A:A.19]

- A.32 **Operational scenario:** Step-by-step description of how the proposed system should operate and interact with its users and its external interfaces (e.g., other systems). Scenarios should be described in a manner that will allow engineers to walk through them and gain an understanding of how all the various parts of the proposed system function and interact as well as validate that the system will satisfy the user's needs and expectations. Operational scenarios should be described for all operational modes, mission phases (e.g., installation, startup, typical examples of normal and contingency operations, shutdown, maintenance, and safing), and critical sequences of activities for all classes of users identified. Each scenario should include events, actions, stimuli, information, and interactions as appropriate to provide a comprehensive understanding of the operational aspects of the system. [Based upon IEEE 1362-1998] Operational scenarios should span all the following items (during nominal, off-nominal, and stressful conditions) that could occur during operations: mission phase, mode, and state transitions; first-time events; operational performance limits; fault protection routines; failure detection, isolation, and recovery logic; operational responses to transient or off-nominal sensor signals; ground-to-spacecraft uplink and downlink. [Based on draft NPR2820]
- A.33 **Operational Software:** Software that has been accepted and deployed, delivered to its customer, or is deployed in its intended environment. [NPR 7150.2A:A.20]
- A.34 **Primary Mission Objectives:** Outcomes expected to be accomplished which are closely associated with the reason the mission was proposed, funded, developed, and operated (e.g., objectives related to top-level requirements or their flow down). [NPR 7150.2A:A.21]
- A.35 **Process Asset Library:** A collection of process asset holdings that can be used by an organization or project. (Definition from source document: CMMI® for- Systems Engineering/Software Engineering/Integrated Product and Process Development Supplier Sourcing.) [NPR 7150.2A:A.22]
- A.36 **Program:** A strategic investment by a Mission Directorate or Mission Support Office that has a defined architecture and/or technical approach, requirements, funding level, and a management structure that initiates and directs one or more projects. A program defines a strategic direction that the Agency has identified as critical. [NPR 7150.2A:A.23]
- A.37 **Project:** A specific investment having defined goals, objectives, requirements, life-cycle cost, a beginning, and an end. A project yields new or revised products or services that directly address NASA's strategic needs. They may be performed wholly in-house; by Government, industry, academia partnerships, or through contracts with private industry. [NPR 7150.2A:A.24] (Software development, maintenance, operations, retirement, management, acquisition, assurance activities and services that are performed, created, or acquired by or for LaRC are also referred to as projects.) [LMS]
- A.38 **Qualification testing:** Testing conducted to determine whether a system or component is suitable for operational use. [IEEE 610.12-1990]
- A.39 **Regression testing:** Selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements. [IEEE STD 610.12-1990]
- A.40 **Release:** A particular version of a configuration item that is made available for a specific purpose (for example, test release). [ISO/IEC 12207:2008]
- A.41 **Reuse:** See software reuse. [NPR 7150.2A:A.25]
- A.42 **Risk Management:** Risk management includes Risk-Informed Decision Making and Continuous Risk Management in an integrated framework. This is done to foster proactive risk management, to better inform decision making through better use of risk information, and then to more effectively manage implementation risks by focusing the Continuous Risk Management process on the baseline performance requirements emerging from the Risk-Informed Decision-Making process. Continuous Risk Management is a systematic and iterative process that efficiently identifies, analyzes, plans, tracks, controls, and communicates and documents risks associated with implementation of designs, plans, and processes. (Definitions from source document: NPR 8000.4, Agency Risk Management Procedural Requirements.) [NPR 7150.2A:A.26 modified]
- A.43 **Safety-Critical Software:** See definition in NASA-STD-8719.13, Software Safety Standard. [NPR 7150.2A:A.32]
- A.44 **Scripts:** A sequence of automated computer commands embedded in a program that tells the program to execute a specific procedure (e.g., files with monitoring, logic, or commands used by software to automate a process or procedure). [NPR 7150.2A:A.27]

- A.45 **Sensitivity Analysis:** The study of how the variation in the output of a model can be apportioned to different sources of variation in the model input and parameters. (Definition from source document: NASA-STD-7009, Standard for Models and Simulations.) [NPR 7150.2A:A.28]
- A.46 **Simulation:** The imitation of the characteristics of a system, entity, phenomena, or process using a computational model. (Definition from source document: NASA-STD-7009, Standard for Models and Simulations.) Only for the purpose of this document, the term “simulation” refers to only those simulations which are implemented in software. [NPR 7150.2A:A.29]
- A.47 **Software:** Computer programs, procedures, scripts, rules, and associated documentation and data pertaining to the development and operation of a computer system. Software includes programs and data. This also includes COTS, GOTS, MOTs, reused software, auto generated code, embedded software, firmware, and open source software components. [NPR 7150.2A:A.30]
- A.48 **Software Architecture:** The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the properties of those components, and the relationships between them. The term also refers to documentation of a system’s software architecture. Documenting software architecture facilitates communication between stakeholders, documents early decisions about high-level design, and allows reuse of design components and patterns between projects. [NPR 7150.2A:A.31]
- A.49 **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.) [NPR 7150.2A:A.33]
- A.50 **Software inspections:** See Software Peer Review/Inspection.
- A.51 **Software item:** An aggregation of software, such as a computer program or database, that satisfies an end use function and is designated for purposes of specification, qualification testing, interfacing, configuration management, or other purposes. Software items are selected based on trade-offs among software function, size, host or target computers, developer, support strategy, plans for reuse, criticality, interface considerations, need to be separately documented and controlled, and other factors. A software item is made up of one or more software units. [J-STD-016:3.1.37]
- A.52 **Software Peer Review/Inspection:** A visual examination of a software product to detect and identify software anomalies, including errors and deviations from standards and specifications (IEEE 1028-2008 IEEE Standard for Software Reviews and Audits). Guidelines for software peer reviews/inspections are contained in NASA-STD-2202-93, Software Formal Inspection Standard. [NPR 7150.2A:A.34]
- A.53 **Software Product:** The set of computer programs, procedures, and possibly associated documentation and data. [ISO/IEC 12207:2008] Examples include plans, requirements, design, code, databases, test information, and manuals. [J-STD-016-1995]
- A.54 **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 the product is supplied from one Government project to another Government project. (Definition from source document: NASA-GB-8719.13, NASA Software Safety Guidebook.) [NPR 7150.2A:A.35]
- A.55 **Software unit:** An element in the design of a software item; for example, a major subdivision of a software item, a component of that subdivision, a class, object, module, function, routine, or database. Software units may occur at different levels of a hierarchy and may consist of other software units. Software units in the design may or may not have a one-to-one relationship with the code and data entities (routines, procedures, databases, data files, etc.) that implement them or with the computer files containing those entities. [J-STD-016:3.1.43] A database may be treated as a software item or as a software unit. [J-STD-016:G.2.4(4.1.a)]
- A.56 **Software Validation:** Confirmation that the product, as provided (or as it will be provided), fulfills its intended use. In other words, validation ensures that “you built the right thing.” [NPR 7150.2A:A.36]

- A.57 **Software Verification:** Confirmation that work products properly reflect the requirements specified for them. In other words, verification ensures that “you built it right.” [NPR 7150.2A:A.37]
- A.58 **Special verification methods:** Any special verification methods for the software item, such as special tools, techniques, procedures, facilities, and acceptance limits. [J-STD-016:F.2.2(4)]
- A.59 **Stakeholders:** Individuals that are affected by or in some way accountable for the outcome of the project (may include project members, suppliers, customer/acquirer [SWE102]), end users, and others). [Based on CMMI V1.2]. Stakeholders also include the project manager, senior management, subsystem leads, and software quality assurance. [LMS]
- A.60 **Static Analysis:** The process of evaluating a system or component based on its form, structure, content, or documentation. (Definition from source document: ISO/IEC 24765:2008 systems and software engineering vocabulary.) [NPR 7150.2A:A.38]
- A.61 **Subsystem:** A secondary or subordinate system within a larger system. (Definition from source document: ISO/IEC 24765:2008 systems and software engineering vocabulary.) [NPR 7150.2A:A.39]
- A.62 **System:** The combination of elements that function together to produce the capability required to meet a need. The elements include hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose. (Definition from source document: NPR 7123.1A NASA Systems Engineering Processes and Requirements.) [NPR 7150.2A:A.40]
- A.63 **Test:** The operation of the software item, or a part of the software item, using instrumentation or other special test equipment to collect data for later analysis. [J-STD-016:F.2.2(4)]
- A.64 **Test coverage:** The degree to which a given test or set of tests addresses all specified requirements for a given system or component. [IEEE STD 610.12-1990]
- A.65 **Uncertainty:** (1) The estimated amount or percentage by which an observed or calculated value may differ from the true value. (2) A broad and general term used to describe an imperfect state of knowledge or a variability resulting from a variety of factors including, but not limited to, lack of knowledge, applicability of information, physical variation, randomness or stochastic behavior, indeterminacy, judgment, and approximation (adapted from NPR 8715.3B, NASA General Safety Program Requirements). [NPR 7150.2A:A.41]
- A.66 **Unit:** See Software Unit. [LMS]
- A.67 **Unit testing:** Testing of individual software units or groups of related units. [IEEE STD 610.12-1990]
- A.68 **Verification methods** (also known as **qualification provisions** or **qualification methods**) may include: a) Demonstration: The operation of the software item, or a part of the software item, that relies on observable functional operation not requiring the use of instrumentation, special test equipment, or subsequent analysis. [J-STD-016:F.2.2(4)] b) Test: The operation of the software item, or a part of the software item, using instrumentation or other special test equipment to collect data for later analysis. [J-STD-016:F.2.2(4)] c) Analysis: The processing of accumulated data obtained from other qualification methods. Examples are reduction, interpolation, or extrapolation of test results. [J-STD-016:F.2.2(4)] d) Software Peer Review/Inspection: A visual examination of a software product to detect and identify software anomalies, including errors and deviations from standards and specifications. [IEEE 1028-2008] Guidelines for software peer reviews/inspections are contained in NASA-STD-2202-93, Software Formal Inspection Standard. [NPR 7150.2A: A.34] The visual examination of software item code, documentation, etc. [J-STD-016:F.2.4(4)] The visual examination of system components, documentation, etc. [J-STD-016:F.2.2(4)] e) Special qualification methods: Any special qualification methods for the software item, such as special tools, techniques, procedures, facilities, and acceptance limits. [J-STD-016:F.2.2(4)]
- A.69 **Wrapper:** See “glueware” definition. [NPR 7150.2A:A.43]

**Appendix B. ACRONYMS**

This Appendix of LPR-7150.2 provides the acronyms list for this LPR and all the LMS-CP-7150.x supporting procedures.

ANSI	American National Standards Institute
ASIC	Application Specific Integrated Circuits
CAD/CAM	Computer-Aided Design and Computer-Aided Manufacturing
CDR	Critical Design Review
CE	Chief Engineer
CMMI®	Capability Maturity Model Integration
CMMI-DEV	Capability Maturity Model® Integration® (CMMI®) for Development
CMU	Carnegie Mellon University
COTS	Commercial-Off-The-Shelf
CP	Center Procedure
CPLD	Complex Programmable Logic Device
CSCI	Computer Software Configuration Item
EDL	Entry, Descent, and Landing
EVA	Extra Vehicular Activity
FAR	Federal Acquisition Regulation
FPGA	Field Programmable Gate Arrays
GOTS	Government-Off-The-Shelf
HQ	Headquarters
IEEE	Institute of Electrical and Electronics Engineers
IV&V	Independent Verification and Validation
LaRC	Langley Research Center
LAPACK	Linear Algebra Package
LAPD	Langley Policy Directive
LIDAR	Light Detection and Ranging
LMS	Langley Management System
LMS CP	Langley Management System Center Procedure
LPR	Langley Procedural Requirements
MAB	Mission Assurance Branch
MOTS	Modified-Off-The-Shelf
NA	Not Applicable
NASA	National Aeronautics and Space Administration
NODIS	NASA Online Directives Information System
NPD	NASA Policy Directive
NPR	NASA Procedural Requirements
OCC	Office of Chief Counsel
OCE	Office of Chief Engineer
OTS	Off-the-shelf
PDR	Preliminary Design Review
PLC	Programmable Logic Controllers
RFP	Request for Proposal
R&T	Research and Technology
SBU	Sensitive But Unclassified
SCAMPI	Standard CMMI® Appraisal Method for Process Improvement
SEI	Software Engineering Institute
SEPG	Software Engineering Process Group

SMA	Safety and Mission Assurance
SMAO	Safety and Mission Assurance Office
SMP	Software Management Plan
SoC	System on Chip
SOW	Statement of Work
SWE	Software Engineering
SWG	Software Working Group
TA	Technical Authority
TCP	Transmission Control Protocol
WBS	Work Breakdown Structure

## Appendix C. REFERENCES

Note: NPDs and NPRs are found in the NASA Online Directives Information System (NODIS) at: <http://nodis3.gsfc.nasa.gov/> Standards are found on the NASA Technical Standards Program Web site at: <https://standards.nasa.gov/>

- a. Electronic and Information Technology Accessibility Standards, - <http://www.gpo.gov/fdsys/pkg/FR-2000-12-21/pdf/00-32017.pdf>.
- b. NPD 7120.4, NASA Engineering and Program/Project Management Policy.
- c. NPD 8700.1, NASA Policy for Safety and Mission Success.
- d. NPR 2800.1, Managing Information Technology.
- e. NPR 8705.2, Human-Rating Requirements for Space Systems.
- f. LAPD 2810.1, Security of Information Technology.
- g. LPR 1440.7, Langley Research Center (LaRC) Records Management Procedural Requirements.
- h. LPR 1620.1: Information Security Program Management Procedures and Guidelines.
- i. NASA-STD-2202-93, Software Formal Inspection Standard.
- j. NASA-STD-3000, Volumes I-II, Man-Systems Integration Standards.
- k. NASA-STD-7009, Standard for Models and Simulations.
- l. EIA/IEEE J-STD-016, Standard for Information Technology – Software Life Cycle Processes – Software Development: Acquirer-Supplier Agreement.
- m. IEEE-STD-828 Section 2.3.4: Configuration Evaluation and Reviews for Additional Guidance on Configuration Audits
- n. IEEE-STD-1362-1998: IEEE Guide for Information Technology – System Definition
- o. IEEE 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.
- p. IEEE 1028-2008 IEEE Standard for Software Reviews and Audits.
- q. IEEE 1012-2004 IEEE Standard for Software Verification and Validation.
- r. ISO/IEC 12207:2008 Systems and Software Engineering – Software Life Cycle Processes.
- s. ISO/IEC 24765:2008 Systems and Software Engineering Vocabulary.
- t. NASA-GB-8719.13, NASA Software Safety Guidebook, <https://standards.nasa.gov/documents/nasa>.
- u. NASA-HDBK 8739.23, NASA Complex Electronics Handbook for Assurance Professionals, <http://www.hq.nasa.gov/office/codeq/doctree/NHBK873923.pdf>.
- v. NASA Aerospace Science and Technology Dictionary, <http://www.hq.nasa.gov/office/hqlibrary/aerospacedictionary/508/508index.htm>.
- w. NASA Engineering Network: Software Engineering Community, <https://nen.nasa.gov/web/software> (which contains NASA OCE sponsored software training, the NASA Software Process Asset Library, Software Architectural Review Board (Recommended Contents for Software Architecture Descriptions), etc.).
- x. NASA IV&V Management System, <http://www.nasa.gov/centers/ivv/ims/home/index.html>.
- y. CMU/SEI-2010-TR-033, CMMI® for Development, Version 1.3, <http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm>.
- z. LaRC Measure Definitions, <https://sites-e.larc.nasa.gov/sweng/supporting-products/>
- aa. Project Risk Workbook Template, (Workbook tool covers Project Progress Review, plus risk and issue tracking – developed at NASA LaRC, Simulation Development & Analysis Branch) <https://sites-e.larc.nasa.gov/sweng/supporting-products/>
- bb. SEL-84-101 Manager's Handbook for Software Development (Revision 1), <https://sites-e.larc.nasa.gov/sweng/supporting-products/> (Methods and aids for the management of software development projects - developed at NASA Goddard Space Flight Center)
- cc. SEL-81-305 Recommended Approach to Software Development (Revision 3), <https://sites-e.larc.nasa.gov/sweng/supporting-products/> (Presents guidelines for an organized, disciplined approach to software development - developed at NASA Goddard Space Flight Center)
- dd. Software Measurement Description for NASA Langley Research Center, <https://sites-e.larc.nasa.gov/sweng/supporting-products/>

## Appendix D. NASA-WIDE SOFTWARE CLASSIFICATION DEFINITIONS

The classification definitions contained in this Appendix use the following terms with the meaning provided in Appendix A. Definitions: airborne vehicle, major engineer/research facility, software, subsystem, system. (The classification definitions below are taken from the source document: NPR 7150.2, NASA Software Engineering Requirements – revision A.)

### Class A: Human Rated Space Software Systems

#### Definition:

Human Space Flight Software Systems\*: (ground and flight) developed and/or operated by or for NASA that are needed to perform a primary mission objective of human space flight and directly interacts with human space flight systems. Limited to software required to perform “vehicle, crew, or primary mission function,” as defined by software that is:

1. Required to operate the vehicle or space asset (e.g., spacesuit, rover, or outpost), including commanding of the vehicle or asset, or
2. required to sustain a safe, habitable<sup>1</sup> environment for the crew, or
3. required to achieve the primary mission objectives, or
4. directly prepares resources (e.g., data, fuel, power) that are consumed by the above functions.

\* Includes software involving launch, on orbit, in space, surface operations, and entry, descent, and landing.

<sup>1</sup> Current standards that address habitability and environmental health, including atmospheric composition and pressure, air and water quality and monitoring, acceleration, acoustics, vibration, radiation, thermal environment, combined environmental effects, and human factors, are documented in NASA-STD-3000, Volumes I-II, Man-Systems Integration Standards.

#### Examples:

Examples of Class A software (human rated space flight) include but are not limited to:

During Launch: abort modes and selection; separation control; range safety; crew interface (display and controls); crew escape; critical systems monitoring and control; guidance, navigation, and control; and communication and tracking.

On Orbit/In Space: Extra Vehicular Activity (EVA); control of electrical power; payload control (including suppression of hazardous satellite and device commands); critical systems monitoring and control; guidance, navigation, and control; life support systems; crew escape; rendezvous and docking; failure detection; isolation and recovery; communication and tracking; and mission operations.

On Ground: pre-launch and launch operations; Mission Control Center (and Launch Control Center) front end processors; spacecraft commanding; vehicle processing operations; and re-entry operations; flight dynamics simulators used for ascent abort calls; and launch and flight controller stations for manned space flight.

Entry, Descent and Landing (EDL): command and control; aero-surface control; power; thermal; and fault protection; and communication and tracking.

Surface Operations: planet/lunar surface EVA; and communication and tracking.

#### Exclusions:

Class A does not include:

1. Software which happens to fly in space but is superfluous to mission objectives (e.g., software contained in an iPod carried on board by an astronaut for personal use), or
2. software that exclusively supports aeronautics, Research and Technology, and science conducted without space flight applications, or
3. systems (e.g., simulators, emulators, stimulators, facilities) used to test Class A systems containing software in a development environment.

**Class B: Non-Human Space Rated Software Systems or Large Scale Aeronautics Vehicles****Definition:**

Space Systems\*: Flight and ground software that must perform reliably to accomplish primary mission objectives, or major function(s) in Non-Human Space Rated Systems. Limited to software that is:

1. Required to operate the vehicle or space asset (e.g., orbiter, lander, probe, flyby spacecraft, rover, launch vehicle, or primary instrument), such as commanding of the vehicle or asset, or
2. required to achieve the primary mission objectives, or
3. directly prepares resources (data, fuel, power, etc.) that are consumed by the above functions.

Airborne Vehicles: Large Scale<sup>1</sup> aeronautic vehicles that are NASA unique in which the software:

1. Is integral to the control of an airborne vehicle, or
2. monitors and controls the cabin environment, or
3. monitors and controls the vehicle's emergency systems.

This definition includes software for vehicles classified as "test," "experimental," or "demonstration" that meets the above definition for Class B software. Also included are systems in a test or demonstration where the software's known and scheduled intended use is to be part of a Class A or B software system.

\* Includes software involving launch, on orbit, in space, surface operations, and entry, descent, and landing.

<sup>1</sup> Large-scale (life-cycle cost exceeding \$250M) fully integrated technology development system – see NPR 7120.8, section 5.1.1.1.

**Examples:**

Examples of Class B software includes but are not limited to:

Space, Launch, Ground, EDL, and Surface Systems: propulsion systems; power systems; guidance navigation and control; fault protection; thermal systems; command and control ground systems; planetary/lunar surface operations; hazard prevention; primary instruments; science sequencing engine; simulations which create operational EDL parameters; subsystems that could cause the loss of science return from multiple instruments; flight dynamics and related data; launch and flight controller stations for non-human space flight.

Aeronautics Vehicles (Large Scale NASA Unique): guidance, navigation, and control; flight management systems; autopilot; propulsion systems; power systems; emergency systems (e.g., fire suppression systems, emergency egress systems; emergency oxygen supply systems, traffic/ground collision avoidance system); and cabin pressure and temperature control.

**Exclusions:**

Class B does not include:

1. Software that exclusively supports non-primary instruments on Non-Human Space Rated Systems (e.g., low cost non-primary university supplied instruments) or
2. systems (e.g., simulators emulators, stimulators, facilities) used in testing Class B systems containing software in a development environment.

**Class C: Mission Support Software or Aeronautic Vehicles, or Major Engineering/Research Facility Software****Definition:**Space Systems:

1. Flight or ground software that is necessary for the science return from a single (non-primary) instrument, or
2. flight or ground software that is used to analyze or process mission data, or
3. other software for which a defect could adversely impact attainment of some secondary mission objectives or cause operational problems, or
4. software used for the testing of space assets, or
5. software used to verify system requirements of space assets by analysis, or
6. software for space flight operations, that is not covered by Class A or B.

Airborne Vehicles:

Systems for non-large scale aeronautic vehicles in which the software:

1. is integral to the control of an airborne vehicle, or
2. monitors and controls the cabin environment, or
3. monitors and controls the vehicle's emergency system.

Systems on an airborne vehicle (including large scale vehicles) that acquire, store, or transmit the official record copy of flight or test data.

Major Engineering/Research Facility: Systems that operate a major facility for research, development, test, or evaluation (e.g., facility controls and monitoring, systems that operate facility-owned instruments, apparatus, and data acquisition equipment).

**Examples:** Examples of Class C software include but are not limited to:

Space Systems: software that supports prelaunch integration and test; mission data processing and analysis; analysis software used in trend analysis and calibration of flight engineering parameters; primary/major science data collection storage and distribution systems (e.g., Distributed Active Archive Centers); simulators, emulators, stimulators, or facilities used to test Class A, B, or C software in a development, integration and test environments (development environment includes environments used from unit testing through validation testing); software used to verify system-level requirements associated with Class A, B, or C software by analysis (e.g., guidance, navigation and controls (GN&C) system performance verification by analysis); simulators used for mission training; software employed by network operations and control (which is redundant with systems used at tracking complexes); command and control of non-primary instruments; ground mission support software used for secondary mission objectives, real-time analysis, and planning (e.g., monitoring, consumables analysis, mission planning).

Aeronautics Vehicles: guidance, navigation, and control; flight management systems; autopilot; propulsion systems; power systems; emergency systems (e.g., fire suppression systems, emergency egress systems, emergency oxygen supply systems, traffic/ground collision avoidance system); cabin pressure and temperature control; in-flight telescope control software; aviation data integration systems; and automated flight planning systems.

Major Engineering/Research Facility: major Center facilities; data acquisition and control systems for wind tunnels, vacuum chambers, and rocket engine test stands; ground-based software used to operate a major facility telescope; and major aeronautic applications facilities (e.g., air traffic management systems; high fidelity motion based simulators).

**Exclusions:** Systems unique to a research, development, test, or evaluation activity in a Major Engineering/Research Facility or Airborne Vehicle where the system is not part of the facility or vehicle and does not impact the operation of the facility or vehicle.

**Class D: Basic Science/Engineering Design and Research and Technology Software****Definition:**Basic Science/Engineering Design:

1. Ground software that performs secondary science data analysis, or
2. ground software tools that support engineering development, or
3. ground software used in testing other Class D software systems, or
4. ground software tools that support mission planning or formulation, or
5. ground software that operates a research, development, test, or evaluation laboratory (i.e., not a Major Engineering/Research Facility), or
6. ground software that provides decision support for non-mission critical situations.

Airborne Vehicle Systems:

1. Software whose anomalous behavior would cause or contribute to a failure of system function resulting in a minor failure condition for the airborne vehicle (e.g., the Software Considerations in Airborne System and Equipment Certification, DO-178B, "Class D"), or
2. software whose anomalous behavior would cause or contribute to a failure of system function with no effect on airborne vehicle operational capability or pilot workload (e.g., the Software Considerations in Airborne System and Equipment Certification, DO-178B, "Class E"), or
3. ground software tools that perform research associated with airborne vehicles or systems.

Major Engineering/Research Facility Related: research software that executes in a Major Engineering/Research Facility but is independent of the operation of the facility.

**Examples:**

Examples of Class D software includes but are not limited to:

Basic Science and Engineering Design: engineering design and modeling tools (e.g., Computer-Aided Design and Computer-Aided Manufacturing (CAD/CAM), thermal/structural analysis tools); project assurance databases (e.g., problem reporting, analysis, and corrective action system, requirements management databases); propulsion integrated design tools; integrated build management systems; inventory management tools; probabilistic engineering analysis tools; test stand data analysis tools; test stand engineering support tools; experimental flight displays evaluated in a flight simulator; and tools used to develop design reference missions to support early mission planning.

Airborne Vehicles: software tools for designing advanced human-automation systems; experimental synthetic-vision display; and cloud-aerosol Light Detection and Ranging (LIDAR) installed on an aeronautics vehicle.

**Exclusions:**

Class D does not include:

1. Software that can impact primary or secondary mission objectives or cause loss of data that is generated by space systems, or
2. software which operates a Major Engineering/Research Facility, or
3. software which operates an airborne vehicle, or
4. space flight software.

**Class E: Small Light Weight Design Concept and Research and Technology Software****Definition:**

1. Software developed to explore a design concept or hypothesis, but not used to make decisions for an operational Class A, B, or C system or to-be built Class A, B, or C system, or
2. software used to perform minor desktop analysis of science or experimental data.

**Examples:** Examples of Class E software include but are not limited to:

parametric models to estimate performance or other attributes of design concepts; software to explore correlations between data sets; line of code counters; file format converters; and document template builders.

**Exclusions:**

Class E does not include:

1. Space flight systems, or
2. software developed by or for NASA to directly support an operational system (e.g., human rated space system, robotics spacecraft, space instrument, airborne vehicle, major engineering/research facility, mission support facility, primary/major science data collection storage and distribution systems), or
3. software developed by or for NASA to be flight qualified to support an operational system, or
4. software that directly affects primary or secondary mission objectives, or
5. software that can adversely affect the integrity of engineering/scientific artifacts, or
6. software used in technical decisions concerning operational systems, or
7. software that has an impact on operational vehicles.

## Appendix E. COMPLIANCE WITH LAWS, POLICIES, REQUIREMENTS

### E.1 Determine the applicable requirements from the below software laws, policies, NPDs, and NPRs:

- a. Ensure that software invention requirements of NPD 2091.1, Inventions Made by Government Employees, are implemented by the project. [NPR 7150.2A:2.1.1]

Note: For more information, contact the Intellectual Property Law Team in the LaRC Office of Chief Counsel (OCC). As early as possible, submit an Invention Disclosure for the software to the OCC to facilitate its review for commercialization and/or release.

- b. Ensure that the project implements software release requirements of NPR 2210.1, Release of NASA Software, by following LMS-CP-1723. [NPR 7150.2A:2.1.3]

Note: It is recommended that if you ever intend to release the software for use outside your organization, you should consult the LaRC Software Releasing Authority during software planning. Contact the Software Releasing Authority in the Information Management Branch for more information.

- c. Ensure that the software technology transfer requirements of NPR 2190.1, NASA Export Control Program, are implemented by the project. Ensure that there will be no access by foreign persons and neither export nor transfer to foreign persons or destinations until an export control review is completed and access/release is approved in accordance with NPR 2190.1, NASA Export Control Program, and NPR 2210.1, Release of NASA Software. [NPR 7150.2A:2.1.2]

Note: Contact the Center Export Administrator for more information.

- d. Ensure that the information security requirements of NPD 2810.1, NASA Information Security Policy, (including means for protecting personal privacy and proprietary information) are implemented by the project. [NPR 7150.2A:2.1.4]

Note: You may also reference LPR 1620.1, Information Security Program Management Procedures and Guidelines, and LAPD 2810.1, Security of Information Technology.

- (1) Consult your Organization Computer Security Official (who is responsible for the organization's Information Technology Security Plan) to determine any project-unique security requirements that need to be fulfilled. [SWE-102.m]

- e. Ensure that the software is accessible to individuals with disabilities in accordance with 36 CFR Part 1194, Electronic and Information Technology Accessibility Standards. [NPR 7150.2A:2.1.5]

Note: For guidance see URL: <http://www.hq.nasa.gov/webaccess/AccessibilityBestPractice.htm>

- f. Ensure that for software acquisitions, developments, or modifications/improvements the LaRC owner of NF 1739, Alternative Future Use Questionnaire – NASA Projects, is contacted to see if this form should be filled out for the project. To determine the owner, go to the Langley Management System at Web site: <https://lms.larc.nasa.gov/index.cfm>, select "Forms," type 1739 in the "Form Number" box, click search, and select "More Details" to see the name of the owner.

[NPR 7150.2A:2.1.6]

## Appendix F. TECHNICAL AUTHORITY MAPPING MATRIX

The majority of this LPR's requirements and supporting LMS CPs have a reference of the form "[SWE-XXX]" appended to them. That reference refers to the parent NPR 7150.2 (revision A) unique Software Engineering (SWE) requirement identifier. The matrix below lists the section title from NPR 7150.2, a title for each NPR 7150.2 requirement, the NPR 7150.2 unique SWE requirement numbers (in the third column) and shows the Technical Authority delegated to each SWE requirement with the authority to approve tailoring/waivers (in the last column). SWE numbers requiring HQ Technical Authority approval are **bold** and **highlighted in yellow**.

[NPR 7150.2A: Appendix D]

Section of NPR 7150.2	Requirement Title*	SWE #	Designated Technical Authority for LPR 7150.2 (and supporting LMS CPs) by Requirement**
<b>Preface</b>	Effective date	1	<b>NASA HQ Chief Engineer</b>
<b>Organizational Capability</b>	Center plan	3	<b>NASA HQ Chief Engineer</b>
	Benchmark	4	<b>NASA HQ Chief Engineer</b>
	Software processes	5	<b>NASA HQ Chief Engineer</b>
	List of agency's programs & projects containing software	6	<b>NASA HQ Chief Engineer/ NASA HQ Chief, Safety and Mission Assurance</b>
<b>Software Life Cycle Planning</b>	Software plans	13	<b>NASA HQ Chief Engineer</b>
	Execute planning	14	Software Manager's Directorate Head
	Cost estimation	15	Software Manager's Directorate Head
	Schedule	16	Software Manager's Directorate Head
	Training	17	Software Manager's Directorate Head
	Reviews	18	Software Manager's Directorate Head
	Software development life cycle or model	19	Software Manager's Directorate Head
	Software classification	20	<b>NASA HQ Chief Engineer</b>
	Software classification changes	21	Software Manager's Directorate Head
	Software assurance	22	<b>NASA HQ Chief Engineer/ NASA HQ Chief, Safety and Mission Assurance</b>
	Software safety	23	<b>NASA HQ Chief Engineer/ NASA HQ Chief, Safety and Mission Assurance</b>
	Plan tracking	24	Software Manager's Directorate Head
	Corrective action	25	Software Manager's Directorate Head
	Changes	26	Software Manager's Directorate Head
<b>Off-the-Shelf (OTS) Software</b>	COTS, GOTS, MOTS, etc.	27	Software Manager's Directorate Head
<b>Verification &amp; Validation</b>	Verification planning	28	Software Manager's Directorate Head
	Validation planning	29	Software Manager's Directorate Head
	Verification results	30	Software Manager's Directorate Head
	Validation results	31	Software Manager's Directorate Head
<b>Project Formulation</b>	CMMI levels for class A, B, and C software	32	<b>NASA HQ Chief Engineer</b>
	Acquisition Assessment	33	Software Manager's Directorate Head
	Acceptance criteria	34	Software Manager's Directorate Head
	Supplier selection	35	Software Manager's Directorate Head
	Software processes	36	Software Manager's Directorate Head
	Software Milestones	37	Software Manager's Directorate Head
	Acquisition planning	38	Software Manager's Directorate Head
<b>Government Insight</b>	Insight into software activities	39	Software Manager's Directorate Head
	Access to software products	40	Software Manager's Directorate Head
	Open source notification	41	Software Manager's Directorate Head
	Electronic access to source code	42	Software Manager's Directorate Head
<b>Supplier Monitoring</b>	Track change request	43	Software Manager's Directorate Head
	Software measurement data	44	Software Manager's Directorate Head
	Joint audits	45	Software Manager's Directorate Head
	Software schedule	46	Software Manager's Directorate Head
	Traceability data	47	Software Manager's Directorate Head
	Solicitation	48	Software Manager's Directorate Head

Section of NPR 7150.2	Requirement Title*	SWE #	Designated Technical Authority for LPR 7150.2 (and supporting LMS CPs) by Requirement**
Software Requirements Development	Document	49	Software Manager's Directorate Head
	Software requirements	50	Software Manager's Directorate Head
	Flow-down & derived requirements	51	Software Manager's Directorate Head
	Bidirectional traceability	52	Software Manager's Directorate Head
	Manage requirements change	53	Software Manager's Directorate Head
	Corrective action	54	Software Manager's Directorate Head
	Requirements validation	55	Software Manager's Directorate Head
Software Design	Document design	56	Software Manager's Directorate Head
	Software architecture	57	Software Manager's Directorate Head
	Detailed design	58	Software Manager's Directorate Head
	Bidirectional traceability	59	Software Manager's Directorate Head
Software Implementation	Design into code	60	Software Manager's Directorate Head
	Coding standards	61	Software Manager's Directorate Head
	Unit test	62	Software Manager's Directorate Head
	Version description	63	Software Manager's Directorate Head
	Bidirectional traceability	64	Software Manager's Directorate Head
Software Testing	Plan, procedures, reports	65	Software Manager's Directorate Head
	Perform testing	66	Software Manager's Directorate Head
	Verify implementation	67	Software Manager's Directorate Head
	Evaluate test results	68	Software Manager's Directorate Head
	Document defects & track	69	Software Manager's Directorate Head
	Models, simulations, tools	70	Software Manager's Directorate Head
	Update plans & procedures	71	Software Manager's Directorate Head
	Bidirectional traceability	72	Software Manager's Directorate Head
	Platform or hi-fidelity simulations	73	Software Manager's Directorate Head
Software Operations, Maintenance, and Retirement	Document maintenance plan	74	Software Manager's Directorate Head
	Plan operations, maintenance & retirement	75	Software Manager's Directorate Head
	Implement plans	76	Software Manager's Directorate Head
	Deliver software products	77	Software Manager's Directorate Head
	As-built documentation	78	Software Manager's Directorate Head
Software Configuration Management	Develop configuration management plan	79	Software Manager's Directorate Head
	Track & evaluate changes	80	Software Manager's Directorate Head
	Identify software configuration items	81	Software Manager's Directorate Head
	Authorizing changes	82	Software Manager's Directorate Head
	Maintain records	83	Software Manager's Directorate Head
	Configuration audits	84	Software Manager's Directorate Head
	Implement procedures	85	Software Manager's Directorate Head
Risk Management	Continuous risk management	86	<b>NASA HQ Chief Engineer/NASA HQ Chief, Safety and Mission Assurance</b>
Software Peer Reviews/ Inspections	Requirements, test plans, design & code	87	Software Manager's Directorate Head
	Checklist, criteria & tracking	88	Software Manager's Directorate Head
	Basic measurements	89	Software Manager's Directorate Head
Software Measurement	Objectives	90	Software Manager's Directorate Head
	Software measurement areas	91	Software Manager's Directorate Head
	Collection & storage	92	Software Manager's Directorate Head
	Analyze data	93	Software Manager's Directorate Head
	Report analysis	94	Software Manager's Directorate Head
Best Practices	Agency process asset library	98	<b>NASA HQ Chief Engineer</b>
	Identify applicable practices	99	<b>NASA HQ Chief Engineer</b>
Training	Software engineering training	100	<b>NASA HQ Chief Engineer</b>
	Software training plan	101	<b>NASA HQ Chief Engineer</b>

Section of NPR 7150.2	Requirement Title*	SWE #	Designated Technical Authority for LPR 7150.2 (and supporting LMS CPs) by Requirement**
<b>Software Documentation Requirements</b>	Software development/management plan	102	Software Manager's Directorate Head
	Software configuration management plan	103	Software Manager's Directorate Head
	Software test plan	104	Software Manager's Directorate Head
	Software maintenance plan	105	Software Manager's Directorate Head
	Software assurance plan	106	NASA HQ Chief Engineer/ NASA HQ Chief, Safety and Mission Assurance
	Center software training plan	107	NASA HQ Chief Engineer
	Center software engineering improvement plan	108	NASA HQ Chief Engineer
	Software requirements specification	109	Software Manager's Directorate Head
	Software data dictionary	110	Software Manager's Directorate Head
	Software design description	111	Software Manager's Directorate Head
	Interface design description	112	Software Manager's Directorate Head
	Software change request/ problem report	113	Software Manager's Directorate Head
	Software test procedures	114	Software Manager's Directorate Head
	Software users manual	115	Software Manager's Directorate Head
	Software version description	116	Software Manager's Directorate Head
	Software metrics report	117	Software Manager's Directorate Head
	Software test report	118	Software Manager's Directorate Head
	Software inspection/peer review/ inspections	119	Software Manager's Directorate Head
<b>Tailoring of Requirements</b>	Submit generic waiver request	120	NASA HQ Chief Engineer
	Document approved alternate requirements	121	NASA HQ Chief Engineer
<b>Designation of Engineering Technical Authority</b>	Center-level Engineering Technical Authority approval	122	NASA HQ Chief Engineer
<b>Compliance</b>	Direction for Technical Authority	124	NASA HQ Chief Engineer
	Compliance matrix	125	NASA HQ Chief Engineer
	Considerations for waivers	126	NASA HQ Chief Engineer
	Review of "P(Center)"	127	NASA HQ Chief Engineer
	Compliance records	128	NASA HQ Chief Engineer
	Check compliance	129	NASA HQ Chief Engineer
<b>Software Life Cycle Planning</b>	Software safety plan	130	NASA HQ Chief Engineer/NASA HQ Chief, Safety and Mission Assurance
	IV&V Plan	131	NASA HQ Chief Engineer/NASA HQ Chief, Safety and Mission Assurance
	Independent Software Classification Assessment	132	NASA HQ Chief Engineer/NASA HQ Chief, Safety and Mission Assurance
	Software safety determination	133	NASA HQ Chief Engineer/ NASA HQ Chief, Safety and Mission Assurance
	Safety-critical software requirements	134	Software Manager's Directorate Head jointly with the Director, SMA at LaRC
<b>Software Implementation</b>	Static analysis	135	Software Manager's Directorate Head
	Validation of software development tools	136	Software Manager's Directorate Head
<b>Software Peer Reviews/ Inspections</b>	Peer Review/inspections of Software plans	137	Software Manager's Directorate Head
<b>Software Documentation Requirements</b>	Software safety plan contents	138	NASA HQ Chief Engineer/NASA HQ Chief, Safety and Mission Assurance
<b>Compliance</b>	"Shall" statements in this NPR	139	NASA HQ Chief Engineer

\* See LPR 7150.2 and supporting LMS CPs for the full requirement description.

\*\* NASA HQ Chief, Safety and Mission Assurance has co-approval on any tailoring/waiver decided at the HQ level that involves safety-critical software. NASA HQ Chief Medical Officer has co-approval on any tailoring/waiver decided at the HQ level that involves software with health and medical implications. Tailoring/waivers decided at the Center level are to follow similar protocol when software safety-critical or health and medical issues are involved. [NPR 7150.2A:Appendix D, Note 3]