

- Objectives**
- To provide a Langley Management System (LMS) Center Procedure (CP) that complies with NPR 7150.2, NASA Software Engineering Requirements, for non-safety-critical Class D software development, maintenance, operations, retirement, management, acquisition, and assurance activities. [SWE-001] [SWE-005] [NPR 7150.2A:6.3.5]
 - To develop the plans for managing the software activities and obtain commitments to the plans.
 - To monitor performance of software activities against the plans.
 - To develop and provide quality software products and services and ensure customer requirements are met.

ApprovalStephen G. Jurczyk, Deputy Director *Original signed on file*Date: May 14, 2013**Scope**

This Center Procedure applies to all non-safety-critical Class D software development, maintenance, operations, retirement, management, acquisition, assurance activities and services that are performed, created, and acquired by or for Langley (hereafter referred to as projects). [SWE-001] [NPR 7150.2A:P.2.1] This includes new software development and modifications to existing software. This procedure applies to both the non-safety-critical Class D software products and associated data. This procedure includes the pretailored requirements from NASA-STD-8739.8, Software Assurance Standard for Class D. [SWE-022]

Compliance Requirements

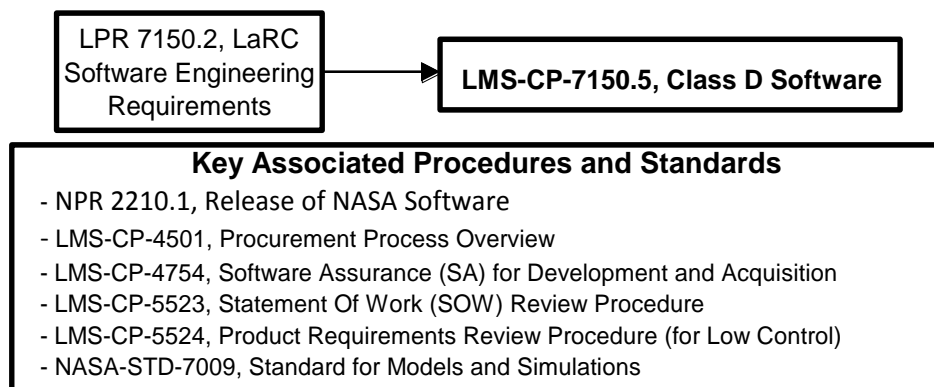
- Text in black is required; software activities shall fully comply with statements in black text. [SWE-139]
- Text in gray is contextual information that provides further description or examples that help clarify the requirement. Gray text is not required to be complied with.
- References are enclosed in brackets “[]”. The references show traceability to source documents.
- References of the form “[SWE-XXX]” refer to requirements numbers assigned to specific requirements in NPR 7150.2, NASA Software Engineering Requirements.

Training & Assistance

For answers to questions or to request training on this procedure, send an email to: 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/ .

Procedure Context Diagram

The context diagram below shows software engineering-related LMS procedures and NASA standards. This procedure is invoked by executing LPR 7150.2, LaRC Software Engineering Requirements. Referenced procedures and standards are performed in parallel with this procedure as required. Definitions, references, and acronyms are located in LPR 7150.2.

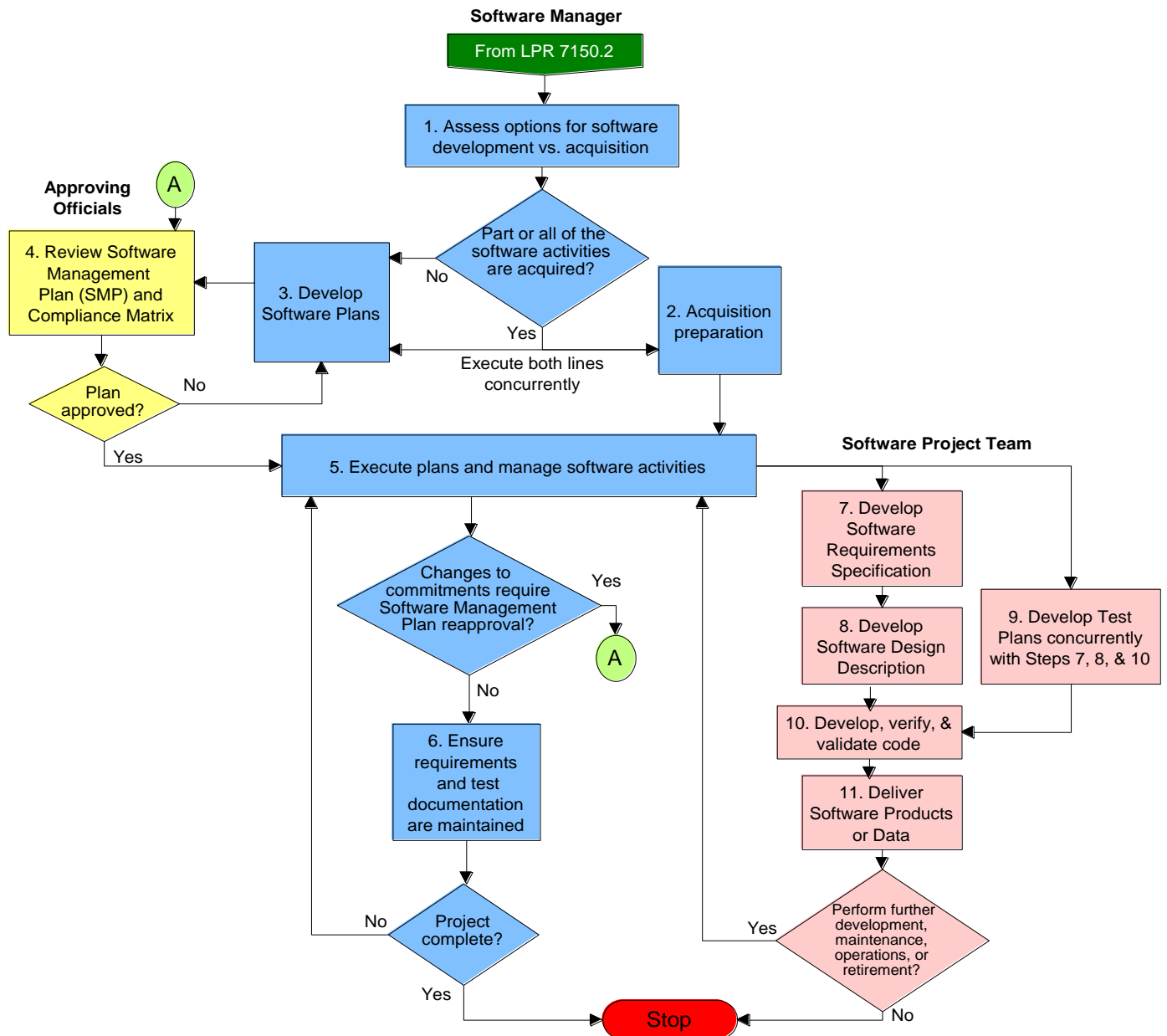
**Records List**

- Software acquisition products (e.g., solicitation, contractual agreement, grant; supplier notification of inclusion of commercial-off-the-shelf (COTS), government-off-the-shelf (GOTS), modified-off-the-shelf (MOTS), reused, or open-source software; supplier-provided deliverables/information called out in the solicitation, contractual agreement, or grant)
- Software life-cycle products (e.g., Software Management Plan (SMP), Software Configuration Management Plan, Software Requirements Specification, Software Design Description, Software Test Plan, Software test results and evaluations, Software Version Description, Software (code and data), records generated by referenced procedures)
- Software product control documentation (e.g., tracking records (such as tracking of defects); and records of approvals)

Table of Contents	Section 1: Perform Software Activities.....	3
	Appendix A: Software Documentation Requirements.....	9
	Appendix A1: Software Management Plan (SMP)	10
	Appendix A2: Software Configuration Management Plan.....	13
	Appendix A3: Software Test Plan	14
	Appendix A4: Software Requirements Specification.....	15
	Appendix A5: Software Design Description	17
	Appendix A6: Software Version Description	18
	Appendix B: Plan Exemptions	19
	Appendix C: LaRC Compliance Matrix for Class D Software (not Safety Critical)	20

Section 1: Perform Software Activities

- Perform the actions defined in the below flow diagram and associated step/action table on the following pages.
- Actions in the step/action table are broken into subactions; both the actions and subactions specified in the step/action table are required to be performed.
- **The steps in the flow diagram are not always performed in the order in which they are numbered.** For example, some requirements analysis and documentation (covered in Step 7) may need to be done before Steps 1, 2, and 3 can be complete. **Review the flow diagram steps to determine if the order is appropriate for the project. Rearrange as necessary (e.g., the steps may be rearranged to accommodate spiral, evolutionary, agile, etc., approaches).** The Software Manager may need to gather applicable project-level documents to prepare for execution of the actions defined in the flow diagram (e.g., Parent Program/Project requirements and plans, science and engineering technical requirements, concept of operation documents).
- Note that many of the activities in the step/action table may be flowed down to a supplier through contractual agreements or grants as part of Step 2.
- If part or all of the software activities are to be acquired, Step 2 is performed in parallel with Step 3.
- The actions defined within each of the individual steps are often performed in parallel and iteratively.
- The actions defined in the flow diagram are performed until the completion or cancellation of the software activity/project.



STEP	ACTION TO TAKE						
1	<p>Assess options for software development vs. acquisition</p> <p>a. Perform any requirements analysis necessary to make an informed make/buy decision by following the actions defined in Step 7. [SWE-050]</p> <p>b. Assess options for software development versus acquisition (make versus buy assessment) and document the resulting acquisition decision. [SWE-033][SWE-038] (See Software Management Plan (SMP), Appendix A1.g.) Guidance: Assess the risk, cost, and benefits of each of the options listed below: [NPR 7150.2A:2.5.2 modified]</p> <p>a. Acquire an off-the-shelf (OTS) software product that satisfies the requirement.</p> <p>b. Develop the software product or obtain the software service internally.</p> <p>c. Develop the software product or obtain the software service through contractual agreement. Note: NPR 2210.1, Release of NASA Software, states that: “generally, NASA does not use grants to develop software.”</p> <p>d. Enhance an existing software product or service.</p> <table border="1" data-bbox="191 600 1516 898"> <thead> <tr> <th data-bbox="191 600 1003 646">If</th> <th data-bbox="1003 600 1516 646">Then</th> </tr> </thead> <tbody> <tr> <td data-bbox="191 646 1003 804">The only acquisition is the acquiring of a commercial-off-the-shelf (COTS), government-off-the-shelf (GOTS), modified-off-the-shelf (MOTS), reused, or open-source software product to be included within a NASA system/subsystem application, and the rest of the software activities are to be performed internally (in-house),</td> <td data-bbox="1003 646 1516 804"> <ul style="list-style-type: none"> • Perform Steps 2.h and 2.i, • Follow LMS-CP-4501, Procurement Process Overview, to procure OTS software products, and • Perform Step 3. </td> </tr> <tr> <td data-bbox="191 804 1003 898">Part or all of the software activities are to be acquired (e.g., develop/modify/enhance a software product or obtain a software service),</td> <td data-bbox="1003 804 1516 898"> <ul style="list-style-type: none"> • Perform Step 2 and Step 3 of this procedure in parallel. </td> </tr> </tbody> </table>	If	Then	The only acquisition is the acquiring of a commercial-off-the-shelf (COTS), government-off-the-shelf (GOTS), modified-off-the-shelf (MOTS), reused, or open-source software product to be included within a NASA system/subsystem application, and the rest of the software activities are to be performed internally (in-house),	<ul style="list-style-type: none"> • Perform Steps 2.h and 2.i, • Follow LMS-CP-4501, Procurement Process Overview, to procure OTS software products, and • Perform Step 3. 	Part or all of the software activities are to be acquired (e.g., develop/modify/enhance a software product or obtain a software service),	<ul style="list-style-type: none"> • Perform Step 2 and Step 3 of this procedure in parallel.
If	Then						
The only acquisition is the acquiring of a commercial-off-the-shelf (COTS), government-off-the-shelf (GOTS), modified-off-the-shelf (MOTS), reused, or open-source software product to be included within a NASA system/subsystem application, and the rest of the software activities are to be performed internally (in-house),	<ul style="list-style-type: none"> • Perform Steps 2.h and 2.i, • Follow LMS-CP-4501, Procurement Process Overview, to procure OTS software products, and • Perform Step 3. 						
Part or all of the software activities are to be acquired (e.g., develop/modify/enhance a software product or obtain a software service),	<ul style="list-style-type: none"> • Perform Step 2 and Step 3 of this procedure in parallel. 						
2	<p>Acquisition preparation</p> <p>a. Develop any technical requirements that need to be flowed down to the solicitation, contractual agreement, or grant by following the actions defined in Step 7. [SWE-050] [SWE-049]</p> <p>b. To put in place a new contractual agreement, follow LMS-CP-4501, Procurement Process Overview, for software supplier selection and establishing the associated proposal evaluation criteria. [LMS]</p> <p>c. Require the software supplier(s)/prospective offerer to notify NASA, in the response to the solicitation, contractual agreement, or grant, as to whether COTS, GOTS, MOTS, reused, or open-source software will be included in code developed for the project. [LaRC Office of Chief Counsel] Guidance: Planning for the inclusion of off-the-shelf software is covered in the SMP, Appendix A1.p, Off-the-shelf (OTS) software.</p> <p>d. Document the NASA software acquisition planning decisions; at a minimum determine the reviews that will be held of the supplier’s progress (e.g., document which supplier products or processes will be reviewed or monitored). [SWE-038] [SWE-039] Guidance: Examples: review of managerial documents such as monthly financial reports and schedule updates; and review of technical products such as designs, test plans, and prototypes. This information may be documented in the SMP, Appendix A1.g, or in an acquisition plan.</p> <p>e. As Step 2 is being performed, document in the Appendix C: Compliance Matrix, all the requirements from this LMS CP that the supplier will perform. [SWE-125] (See SMP, Appendix A1.c, for additional instruction.)</p> <p>f. Document in the solicitation, contractual agreement, or grant the following: [SWE-038] [SWE-048]</p> <p>1. Software processes, activities, and tasks that will be performed by the supplier, and what associated products and documents will be delivered. [SWE-048][SWE-125][SWE-102.e]</p> <p>(a) Determine which requirements in this LMS CP are to be flowed down and accomplished by the supplier and document them in the solicitation, contractual agreement, or grant (i.e., include all processes, activities, and tasks required to be performed by the supplier). [SWE-048] [SWE-038] [NPR 7150.2A:P2.2 Note] [NPR 7150.2A:6.3.5]</p> <p>(b) Record in the deliverables list the applicable products and documents that the supplier is required to provide, including appropriate documentation to support operations and maintenance (e.g., Appendix A1 through Appendix A6; the supplier Software Management Plan, Software Configuration Management Plan, and Software Maintenance Plan; requirements, design, and test documentation; code and data; as well as the Software Version Description, the as-built (up-to-date) documentation, and the Software Users Manual). [SWE-038] [SWE-077] [NPR 7150.2A:P2.2 Note] [NPR 7150.2A:6.3.5]</p>						

STEP	ACTION TO TAKE
	<p>2. Reviews that will be held with the supplier to assess progress and the milestones at which the reviews will be held. [SWE-037] [SWE-102.h] [SWE-102.j] Guidance: Reviews should include both management reviews and technical reviews. Management reviews evaluate project status relative to the applicable project plans and schedules. Technical reviews evaluate the software products or services against criteria such as product completion, compliance with standards and specifications, adherence to Configuration Management Plans, and product readiness. [Based on IEEE 12207.0-1996] Technical reviews may be of preliminary documents in order to flush out details or unknowns, or of final documents to make sure they are correct, complete, etc. (See SMP, Appendix A1.e.2.d.)</p>
	<p>3. Supplier-provided information and data which are to be delivered in order to provide insight. [SWE-039] [SWE-102.h] [SWE-102.j] At a minimum the following must be included:</p>
	<p>(a) The products and data that the project plans to review. [SWE-039]</p>
	<p>(b) The activities, and reviews the project wants to participate in (e.g., activities may include reviews of trade study data and results, peer reviews of the Software Requirements Specification and Software Design Description, monitoring integration, reviews to determine the verification adequacy, participation in software code peer reviews, and participation in systems and software technical interchange meetings). [SWE-039]</p>
	<p>(c) Software metrics (e.g., software progress tracking measures, quality measures, requirements volatility measures). [SWE-044] For example: number of milestones and deliverables completed, cost (planned, actual, and revised estimate to complete), schedule planned and actual, number of test issues open/closed, number of acceptance issues open/closed, number of requirements changes.</p>
	<p>(d) The software schedule for the project's review and the frequency at which the schedule updates will be provided. [SWE-046]</p>
	<p>4. Acceptance criteria (and any necessary conditions under which the criteria must be met) for software. [SWE-034] Guidance: Examples of conditions are operating under maximum loads such as number of simultaneous users or peak bandwidth capacity, and operational modes such as startup, operations, shutdown, and maintenance.</p>
	<p>5. That all software requirements are to be flowed down to all levels of software supplier (e.g., sub-contracts). [SWE-001] [NPR 7150.2A:P.2.2] [NASA-STD-8739.8:6.9.1]</p>
	<p>6. Optional : that the software supplier(s) provides NASA: [SWE-040]</p>
	<p>(a) All software products in electronic format, including: [SWE-040] (1) Software process tracking information, including software development and management metrics, and [SWE-040]</p>
	<p>(2) Software traceability data for the project's review (e.g., software requirements traceability to higher-level requirements and to software tests). [SWE-047]</p>
	<p>(b) Electronic access to the source code developed for the project, including MOTS software. [SWE-042] Guidance: If the project needs to control future use and distribution of the delivered software or requires unlimited rights to the software (e.g., right to use, modify, and distribute the software for any purpose), the project should consider having the software copyright assigned to the Government. The project should coordinate with the Office of Chief Counsel (Patent/Intellectual Property Counsel) and contracting officer regarding required rights associated with the software. [NPR 7150.2A:2.6.1.4 modified] [LaRC Office of Chief Counsel]</p>
	<p>7. Optional: follow-on support if needed such as maintenance and operations support.</p>
	<p>g. Review the Statement Of Work using LMS-CP-5523, Statement Of Work (SOW) Review Procedure. [LMS]</p>
	<p>h. Consult with the Office of Chief Counsel and the contracting officer regarding the inclusion of COTS, GOTS, MOTS, reused, or open-source software within a NASA system or subsystem. [LaRC Office of Chief Counsel]</p>
	<p>i. Additional requirements on the inclusion of open-source software are in NPR 2210.1, Release of NASA Software, Section 1.8.3. (See the NASA Online Directives Information System at http://nodis3.gsfc.nasa.gov/) [LaRC Office of Chief Counsel]</p>
<p>3</p>	<p>Develop software plans a. Perform any requirements analysis needed to make an informed planning decision by following the actions defined in Step 7. [SWE-050]</p>

STEP	ACTION TO TAKE
	<p>b. Ensure that:</p> <ol style="list-style-type: none"> 1. Each system and subsystem containing software has been classified in accordance with the software classification definitions in Appendix D in LPR 7150.2. [SWE-020] 2. The LaRC Mission Assurance Branch has: <ol style="list-style-type: none"> (a) Performed an independent Software Assurance Classification Assessment of each system and subsystem containing software according to LMS-CP-4754, Software Assurance (SA) for Development and Acquisition. [SWE-132] (b) Determined that the software is not safety critical in accordance with LMS-CP-4754. [SWE-133] 3. The classification and safety-critical determination results are recorded in or referenced from the SMP. [SWE-102.b]. (See SMP, Appendix A1.b.) 4. The NASA software inventory data for the project (e.g., software class, safety criticality) is provided to the LaRC current representative to the Agency Software Working Group. (See https://sites-e.larc.nasa.gov/sweng/home_pg/ for name of current representative.) [SWE-006] <p>c. Ensure that the project follows the requirements specified in Appendix A: Software Documentation Requirements, when creating the documents required by this procedure (i.e., Appendix A1 through Appendix A6). [NPR 7150.2A:Chapter 5]</p> <p>d. Develop the following software plans: [SWE-013] See Appendix B: Plan Exemptions, for exemptions pertaining to planning.</p> <ol style="list-style-type: none"> 1. Software Management Plan (SMP) as defined in Appendix A1. [SWE102] [SWE-013] [SWE-125] <ol style="list-style-type: none"> (a) If part of the work is being performed under a contractual agreement or grant: <ol style="list-style-type: none"> (1) NASA civil servants develop an SMP (sometimes called an acquisition plan) to cover the organization structure; software class and safety-criticality determinations; Compliance Matrix; Work Breakdown Structure (WBS); schedule; and activities performed by the civil servants. (2) The supplier also produces an SMP to cover the supplier's organization structure, engineering environment, WBS, schedule, costs, staffing, requirements management process, verification and validation plan, and other activities delegated under the contract or grant. (3) Some overlap will occur, but topics are covered in the NASA SMP from a civil servant acquirer perspective and in the supplier's SMP from a supplier perspective. (b) For projects that involve noncontractual multi-party agreements (support, partnership, and inter-Center agreements), also follow LPR 7150.2, Section 1.4, Multi-Party Software Activities, which specifies the required content for shared agreements involving software activities; the shared agreement content is defined concurrently with the SMP and Compliance Matrix. 2. Software Configuration Management Plan as defined in Appendix A2. [SWE103] [SWE-013] [SWE-079] 3. Optional: Software Maintenance Plan (See LMS-CP-7150.4, Appendix A4, for example Software Maintenance Plan content).
4	<p>Review Software Management Plan (SMP) and Compliance Matrix</p> <ol style="list-style-type: none"> a. The Software Manager reviews and approves the SMP, including the Compliance Matrix and any applicable supporting acquisitions plans, to ensure it is ready for implementation and provides the plan to the Line Manager for approval. [LMS] b. The Software Manager's Line Manager reviews and approves the Compliance Matrix and the SMP to ensure they are in compliance with this procedure, to approve the allocation of the Line Manager's staff and resources (e.g., licenses, travel, etc.), and to approve any tailoring requests in the Compliance Matrix. [LMS] [NASA-STD-8739.8:5.4.1.1] Guidance: Additional approvals may need to be obtained for the SMP (e.g., the Project Manager may need to approve funding and staffing). Note: Line Manager should also check those items marked as "NA" to ensure they are appropriate. The NASA Headquarters Office of the Chief Engineer periodically conducts appraisals to check compliance. c. If the Compliance Matrix contains requests for tailoring against any of the requirements in this LMS CP, the Software Manager follows LPR 7150.2, Section 2.3, to obtain all the required approvals for the tailoring. [SWE102.c]
5	<p>Execute plans and manage software activities</p> <ol style="list-style-type: none"> a. When tailoring or waivers have been approved against the software Compliance Matrix, ensure a copy of the approved matrix is provided to the Software Engineering Process Group (SEPG) via email at LaRC-DL-SW-Matrix@mail.nasa.gov. [SWE-128]

STEP	ACTION TO TAKE
	<p>b. Ensure the implementation/execution and maintenance of the: [SWE-014]</p> <ol style="list-style-type: none"> 1. Software Management Plan (SMP), [SWE-014] [SWE-102] [SWE-085] 2. Software Configuration Management Plan, and [SWE-014] [SWE-103] 3. Optional: Software Maintenance Plan. [SWE-085] <p>c. If a system or subsystem evolves to a higher software classification than Class D, revisit LPR 7150.2 to complete Sections 1.2 and 1.3; this procedure no longer applies. [SWE-021] [SWE-132]</p> <p>d. If at any time the project is determined to have safety-critical software, revisit LPR 7150.2 to complete Sections 1.2 and 1.3; this procedure does not apply to the safety-critical software. [SWE-023] [SWE-133]</p> <p>e. As defined in the SMP, ensure that actual milestones and deliverables are tracked against the software plans. [SWE-024]</p> <p>f. Maintain the Compliance Matrix against requirements in this LMS CP including those delegated to other parties or accomplished by contract vehicles. [SWE-125]</p> <ol style="list-style-type: none"> 1. Ensure that all matrix items are implemented, and [SWE-125] 2. If the project desires to modify the approved Compliance Matrix, follow LPR 7150.2, Section 2.3.7, to obtain approval for the modifications via waiver. [SWE-125] 3. When waivers have been approved against the software Compliance Matrix, ensure a copy of the approved matrix is provided to the Software Engineering Process Group (SEPG) via email at LaRC-DL-SW-Matrix@mail.nasa.gov <p>g. Ensure that changes to commitments (e.g., software plans) are agreed to by the affected groups and individuals.</p> <ol style="list-style-type: none"> 1. If commitments (staff, resources, schedule, deliverables) change, return to the Step 4 to obtain approval. [NASA-STD-8739.8:5.4.1.1] 2. Retain records of approvals (e.g., the Line Manager's approval of the SMP and Compliance Matrix). [SWE-128]
6	<p>Ensure requirements and test documentation are maintained</p> <p>a. Ensure the following are maintained: [SWE-050] [SWE-065] [SWE-072]</p> <ol style="list-style-type: none"> 1. Software Requirements Specification, [SWE-050] 2. Software Test Plan(s), this includes updating the test plan bidirectional traceability to be consistent with changes to the Software Requirements Specification. [SWE-065.a] [SWE-072]
7	<p>Develop Software Requirements Specification</p> <p>a. Identify and develop software requirements based on analysis of: [SWE-050]</p> <ol style="list-style-type: none"> 1. Customer and other stakeholder requirements, and [SWE-050] 2. The operational concepts/operational scenarios (see LPR 7150.2 for the definition of operational scenarios). [SWE-050] Guidance: Early user involvement is fundamental to ensuring that the requirements and the operational concepts accurately reflect user expectations. It is highly recommended that an operational concept be documented, reviewed, and approved to ensure that the requirements adequately cover the agreed upon operational concept and user needs. <p>b. Document the software requirements as defined in Appendix A4: Software Requirements Specification. [SWE-049] [SWE-050] [SWE-109]</p> <ol style="list-style-type: none"> 1. Use Appendix A: Product Requirements Checklist, from LMS-CP-5524, Product Requirements Review Procedure, to prepare and review the Software Requirements Specification and track any defects identified in the review to closure. [SWE-029] [SWE-031] [LMS] <p>c. Obtain approval of the Software Requirements Specification. [SWE-050]</p>
8	<p>Develop Software Design Description</p> <p>a. Transform the requirements into a documented Software Design Description as defined in Appendix A5. [SWE-057] [SWE-111]</p> <p>b. Review the Software Design Description to make sure that it is correct and accurately reflects the Software Requirements Specification and track any defects identified in the review to closure. [SWE-028] [SWE-030] Guidance: For checklists to aid in reviewing designs, see Sections I0 and I1 of the Peer Review Inspection Checklists file at: PeerReviewInspectionChecklistsR2V1.doc .</p>
9	<p>Develop Test Plans concurrently with Steps 7, 8, & 10.</p> <p>a. Document the Software Test Plan as defined in Appendix A3. [SWE-065 thru SWE-065.a] [SWE-104]</p>

STEP	ACTION TO TAKE
10	<p>Develop, verify, and validate code</p> <p>a. Implement the software design into software code. [SWE-060]</p> <p>b. Ensure that the key units of the software code are unit tested. [SWE-062] [SWE-104.b.1] Guidance: Key units may include: units controlling interfaces, units containing complex algorithms, units with timing constraints or that manage control flow.</p> <p>c. Perform software testing as defined in the Appendix A3, Software Test Plan. [SWE-066] [SWE-068] [SWE-069]</p> <p>d. Perform all acceptance activities documented in the SMP. [SWE-031] [SWE-034] [SWE-102.i] (See SMP, Appendix A1.h.)</p>
11	<p>Deliver Software Products or Data</p> <p>a. Provide a Software Version Description as defined in Appendix A6 for each software delivery to the requester or intended user. [SWE-063] [SWE-116] Guidance: A Software Version Description will be produced whenever software is created/modified to produce delivered code or data.</p> <p>b. Complete and deliver the software product to the customer with appropriate documentation to support the operations and maintenance phase of the software's life-cycle. [SWE-077] (See SMP, Appendix A1. e.1.c)</p> <p>1. However, if the code is not a deliverable (i.e., only data or analysis are delivered or published), the project documents and retains the appropriate documentation to build, execute, and recreate the data or analysis. [SWE-077] This documentation may be included in the Software Version Description.</p>

Appendix A: Software Documentation Requirements

- a. The project completes the subappendices (Appendix A1 through A6) of this Appendix.
 1. The subappendices specify the required content of software documents, but the format and style of the documents are left up to the project (e.g., Branch template or Wiki can be used, or the requirements may be embedded in branch procedures such as a Configuration Management Branch Procedure).
 2. As long as the required content is addressed, subappendices can be combined or requirements within a subappendix can be rolled out as a separate document.
 3. If the software activities are part of a larger/parent project, content requirements in Appendix A1 through A6 can be fulfilled by the parent project documentation if the required content is fully addressed.
- b. In Appendix A1 through A6, text in black is required; software activities shall fully comply with statements in black text unless a request for tailoring is documented in the Compliance Matrix (specified in Appendix C) and approved following LPR 7150.2, Section 2. [SWE-139] Text in gray is contextual information that provides further description or examples to help clarify the requirement or additional items (recommended by industry standards or NASA NPR) to be included in the document.
- c. For the reader's convenience in completing the following subappendices, an electronic Microsoft Word version of Appendix A1 through A6 and Appendix C is provided in the "Appendices for LMS-CP-7150.5: Class D Software" at: <https://sites-e.larc.nasa.gov/sweng/supporting-products/> .
- d. The project shall clearly note "Not Applicable" document content requirements as "NA" in both the document and the project's Compliance Matrix (specified in Appendix C); only those requirements marked with an asterisk "*" in Appendix A1 through A6 may be denoted as "NA" without required approval via LPR 7150.2. The omission or modification of document content requirements without an asterisk does require approval following LPR 7150.2, Section 2, Tailoring and Waivers.
 1. Specific content within the subappendices (Appendix A1 through A6) may not be applicable for every project.
 2. The decision(s) to make document content requirement items not applicable (NA) may be reviewed by external organizations (e.g., Mission Assurance Branch, Headquarters Office of Safety and Mission Assurance auditors, and Headquarters Office of Chief Engineer auditors). [NPR 7150.2A:Chapter 5 modified] For the benefit of reviewers, the project should document a short justification for each NA. [HQ OCE]

Appendix A1: Software Management Plan (SMP)

Instructions: The SMP is a living document. Therefore, some of the plan's required content may not be known at the time of its initial release and approval. However, for unknown items, expected closure dates should be assigned and tracked.

The Software Management Plan shall contain: [SWE-102] [SWE-013]

- a. **Project Organization.** Project organizational structure showing authority and responsibility of each organizational unit, including contractual suppliers and external organizations (e.g., universities, other government organizations, and industry partners). [SWE-102.a] [SWE-103.a]
 1. Include the Mission Assurance Branch in the organizational structure because it performs, at a minimum, the Software Assurance Classification Assessment and safety-criticality determination as defined in LMS-CP-4754, Software Assurance (SA) for Development and Acquisition. [SWE-102.a] [LMS]
 2. Include as an external interface to the organizational structure the Technical Authority for approving any project requested software tailoring or waivers. [SWE-102.a] [SWE-122]
- b. **Software Classification and Safety Criticality.** The software classification of each of the systems and subsystems containing software and the safety-criticality determination. (e.g., All the systems containing software on the project are Class D and non-safety critical.) [SWE-102.b] [SWE-020]
- c. **Compliance Matrix.** The completed Appendix C, Compliance Matrix, is included as an appendix to the SMP. [SWE-102.c] [SWE-125] A Microsoft Word version of the Compliance Matrix (as specified in Appendix C) is provided in the "Appendices for LMS-CP-7150.5: Class D Software" at: <https://sites-e.larc.nasa.gov/sweng/supporting-products/>.
 1. For those requirements delegated to other parties (e.g., other Centers, agencies, grantees, or partners) or accomplished by contract vehicles, identify the responsible party that the requirement is delegated to in the Compliance Matrix. [SWE-125] [NPR 7150.2A:P.2.2]
 - (a) For requirements delegated to other parties see further instruction in LPR 7150.2, Section 1.4. [SWE-125]
 2. Document the tailoring requests against requirements in this procedure in the Compliance Matrix; tailoring is defined in LPR 7150.2, Section 2.3.2. [SWE-102.c] [SWE-125 Note] [SWE-128]
 3. Only requirements from Appendix A1 through A6 marked with an asterisk "*" may be denoted as "NA" in the Compliance Matrix. See Appendix A for the restrictions on the use of NA.
- d. **Engineering environment.** [SWE-102.d] Describe software, hardware, and facilities necessary to develop the software products and, as applicable, those items necessary to operate and maintain the software products. This may include computing platform, operating system, libraries (Linear Algebra Package (LAPACK)), equipment (logic analyzer, emulators), standards (Unified Modeling Language Version 2.3, ISO/IEC 14882: Programming Language-C++), procedures (use of software development folders to maintain current development progress), and tools (compiler, debuggers, static analyzers).
- e. **WBS, Schedule, Effort, and Cost.** [SWE-102.e] [SWE-016] [SWE-015]
 1. Document a list (e.g., Work Breakdown Structure (WBS)) of: [SWE-102.e]
 - (a) Software activities (e.g., developing plans, defining the requirements, developing the design, developing the code, implementing the tests, deliver of products/data, perform acceptance activities). [SWE-036] [SWE-102.e]
 - (b) Software products that will be produced (e.g., Software Management Plan, Software Configuration Management Plan, Software Requirements Specification, Software Design Description, Software Test Plan, Software Version Description, software code, software data or analysis results, and Software Users Manuals). [SWE-102.e] [SWE-077]
 - (c) Appropriate software documentation to be produced to support the operations and maintenance phases of the life cycle (e.g., Software Version Description that includes the software build and execution instructions, and the up-to-date versions of the appropriate software products listed above in e.1.b). [SWE-077] [SWE-102.e] The project should also consider developing a Software User Manual for the completed software; see Appendix A16 of LMS-CP-7150.4: Class C Software for suggested content. [NPR 7150.2A:5.2.7.1] The extent of documentation "to support the operations and maintenance" will depend on the items delivered. For example, a delivered executable may require little or no "build" documentation. Documentation can take the form of text, audiovisual instructions, interactive scripts, help files, man pages, embedded instructions (e.g., invoked by passing a "help" argument to the software), etc.
 - (d) *Software services and nondeliverable items to be performed. [SWE-102.e]
 2. Document the software schedule associated with the activities, services, and deliverables that satisfies the following minimum conditions: [SWE-016] Note: the "Wide-Ban Delphi Estimation Process" at: <https://sites-e.larc.nasa.gov/sweng/supporting-products/> provides an example process for estimating size, cost, or other types of estimates.
 - (a) Coordinates with the overall project schedule if the software is an element of a larger project [SWE-016.a] and
 - (b) Documents the schedule for the activities defined in Section 1 of this procedure. [SWE-102.e]

- (c) Documents milestones and delivery dates. [SWE-016.b] [SWE-102.e]
- (d) *As part of acquisitions activities, documents milestones at which the software supplier(s) progress will be reviewed by the project or reference the contractual agreement where the milestones are defined. [SWE-037]
3. Document an estimate of effort (include both civil servant and contractor effort) that covers either: [SWE-015]
- (a) The entire software life cycle, or [SWE-015.a]
- (b) If the software activity is ongoing, then the effort estimate may be for a period of performance (e.g., one year) rather than the whole life cycle. [SWE-015.a]
4. *Document an estimate of special costs that covers the entire software life cycle or period of performance. [SWE-015.a] Special costs do not include labor and can include (but are not limited to) travel, training, and OTS software or hardware purchases and installation fees. It does not include items provided to but not purchased by the project, which may include developer workstations and software development tools.
- f. **Requirements Management.** Describe how the project will collect and manage changes to the software requirements. [SWE-053][SWE-102.g] The project could document requirements changes either by: a) keeping a revision history of the changes to the Software Requirements Specification (e.g., Microsoft Word revision marks), or b) keeping old hand-marked versions, or c) implementing a change request system. Managing changes includes making all stakeholders aware of the changes.
- g. **Contractor management.** Include the following: [SWE-102.h]
1. The make-versus-buy decision (i.e., document the results of Step 1.b in Section 1 of this procedure). [SWE-033] [SWE-102.e]
2. *The acquisition planning decisions and the activities to be performed to gain insight on the supplier (e.g., monthly review of schedules, metrics, financial and reports, etc.). [SWE-038] [SWE-039] [SWE-102.h]
- This information may be included in this plan, or reference an acquisition plan or contractual agreement where it is documented.
- h. **Software acceptance.**
1. Document the acceptance criteria that will be used to confirm that the software fulfills its intended use (and any necessary conditions under which the acceptance criteria must be met). [SWE-029][SWE-034][SWE-102.i] The acceptance criteria and conditions may be documented in this plan, the Software Test Plan (see Appendix A3) and/or in the contractual agreement. Examples of necessary conditions are operating under maximum loads such as maximum number of simultaneous users or peak bandwidth capacity; and operational modes such as startup, operations, shutdown, and maintenance. The following are suggested examples of acceptance criteria and conditions:
- (a) Complete a review of the Software Test Plan to ensure:
- (1) It is up to date with the latest version of the Software Requirements Specification,
- (2) That the plan contains the necessary and sufficient tests to make sure that the product fulfills all requirements in the Software Requirements Specification (and, if applicable, all requirements in the contractual agreement), and
- (3) That the plan covers tests and analysis to make sure that the software's actual output meets the expected output for execution under 1) minimum load/usage, 2) normal load/usage, and 3) maximum load/usage.
- (b) Make sure that a review of the final versions of all the documents has been performed to ensure that they are complete, spell checked, up to date, and accurate.
- (c) The source code compiles on the target with no syntax and no logical errors.
- (d) The software passes all tests in the Software Test Plan on the target computer.
- (e) The users report zero defects in X days of use.
2. Describe how the project will ensure software products meet acceptance criteria and any conditions (e.g., by performing acceptance testing, final customer checkout or readiness review). [SWE-029] [SWE-102i] For example:
- (a) Successfully compile the software on the target computer with no syntax and no logical errors.
- (b) As part of acceptance testing, execute the Software Test Plan on the target computer.
- (c) The customer performs "hands-on" testing by exercising the software as users for a period of X-days prior to final acceptance.
3. Describe how the project will record results of software acceptance, and address, and track issues to closure (e.g., a test log, defect tracking system, or validation matrix). [SWE-031] [SWE-102.i] [SWE-069] For example:
- (a) All identified defects will be documented in the test matrix.
- (b) Personnel will be assigned to:
- (1) Correct each defect and test each correction,
- (2) Update the Software Test Plan to include any new tests that were created to test the corrections, and
- (3) When complete, record the completion in the test matrix along with the version of the software and if changed, the version of the Software Test Plan.
- (c) The customer will witness the final testing.

- (d) Final testing will consist of re-executing the Software Test Plan on the target computer to validate that all identified errors have been shown to be corrected and that no new errors have been introduced by the corrections.
4. * If applicable, describe how the project will perform any additional activities (other than testing) such as demonstration, analysis, or inspection to confirm the product fulfills its intended use and requirements; and how the project will record results of those activities, and address, and track issues to closure. [SWE-102.i] [SWE-028] [SWE-029] [SWE-031] For model and simulation software, information regarding specific verification and validation techniques and the analysis of models and simulations can be found in NASA-STD-7009, Standard for Models and Simulations (e.g., numerical accuracy, uncertainty analysis, and sensitivity analysis, as well as verification and validation for software implementations of models and simulations).
- i. Reviews. [SWE-018]
1. Define the reviews that the project will hold regularly of software activities, status, and results with project stakeholders and the frequency or schedule for the reviews. [SWE-018] [SWE-102.i] Stakeholders are individuals that are affected by or in some way accountable for the outcome of the project (*may include project members, suppliers, customer/acquirer, end users, and others*). [Based on CMMI-DEV, V1.3]
 2. Describe how the project will track issues from reviews to resolution. [SWE-018] [SWE-102.i]
- j. *Approvals. If applicable, any additional approval required by such means as regulations, required certifications, proprietary, usage, ownership, warranty, and licensing rights. [SWE-102.n]
- k. Plan Tracking. Describe how the project will ensure that actual milestones and deliverables are tracked against the SMP. [SWE-024] [SWE-102.o] This may be done as part of performing item i above.
- l. Software development life-cycle phases. [SWE-019] [SWE-102.q] The phases can be included as part of the WBS. The following are typical life-cycle phases: requirements phase, design phase, coding phase, integration phase, test phase, acceptance phase, and maintenance phase. The project may also want to document the software life-cycle model to be used on the project (e.g., waterfall, spiral, agile). For additional guidance on choosing a life cycle, refer to "Guidance on Selecting Life Cycle and Development Approach" at: <https://sites-e.larc.nasa.gov/sweng/supporting-products/>.
- m. Configuration Management. Include here or provide the reference to the Software Configuration Management Plan as defined in Appendix A2. [SWE-102.r]
- n. Maintenance. Describe how the project will maintain software products and data. [SWE-075] [SWE-085] For example: The project will follow Section 1: Perform Software Activities of LMS-CP-7150.5 and the Software Configuration Management Plan to perform maintenance.
- o. *Metrics. If part of the software products or services are being supplied under a contractual agreement, describe how the project will collect and analyze the supplier provided software metrics. [SWE-102.v] [SWE-044] (See Step 2.f.3.c in Section 1 of this procedure for more details.)
- p. Off-the-shelf (OTS) software. Management, development, and testing approach for handling any commercial-off-the-shelf (COTS), government-off-the-shelf (GOTS), modified-off-the-shelf (MOTS), reused, or open-source software component(s) that are included within a NASA system or subsystem. [SWE-102.x] Also, address any proprietary, usage, ownership, warranty, licensing rights, transfer rights and any future support for the software product. [SWE-027.c] [SWE-027.d]
- q. Operations and retirement.
1. Plan for software operations activities. For example: performing specific operations, training operators, or providing support when problems are encountered during operations. [SWE-075]
 2. Plan for software retirement activities. [SWE-075]

Appendix A2: Software Configuration Management Plan

Instructions: Software Configuration Management Plan may be included in the Software Management Plan or rolled out as a separate document.

The Software Configuration Management Plan shall contain: [SWE-013] [SWE-079] [SWE-103]

- a. The project name. [SWE-103.a]
- b. Assigned responsibilities and authority for the implementation of software configuration management on the project. [SWE-103.b] [SWE-079]
- c. References to the software configuration management policies and directives that apply to the project (e.g., this LMS CP and other project or branch configuration management documented requirements). [SWE-103.c]
- d. Describe the functions and tasks required to manage the configuration of the software, including the following: [SWE-079] [SWE-103.d]
 1. Describe how the project will identify and record the software configuration items (e.g., software documents, code, data, tools, models, scripts) and their versions to be controlled for the project, and [SWE-081] [SWE-079] [SWE-103.d]
 2. Describe how the project will track changes to software products. [SWE-079] [SWE-080] [SWE-103.d]
- e. Schedule information, which establishes the sequence and coordination for the identified activities. [SWE-103.e]
- f. Identification of any configuration management tools used. [SWE-103.f]
- g. Plan maintenance information, which identifies the activities and responsibilities necessary to keep the Software Configuration Management Plan up to date. [SWE-103.g] This could be something as simple as stating: "On an as-needed basis, changes to the Software Configuration Management Plan will be made and submitted to the Software Manager for approval."
- h. Define how the project will store, back up, and deliver/release deliverable software products and data. [SWE-085] [SWE-103.h]

Appendix A3: Software Test Plan

The Software Test Plan shall include: [SWE-028] [SWE-013] [SWE-065] [SWE-104] [SWE-114]

- a. Describe how the project will: [SWE-065.a]
 1. Document test inputs, [SWE-114.b.4]
 2. Record and evaluate test results, [SWE-030] [SWE-068] [SWE-069] [SWE-104.f]
 3. Document the evaluation, [SWE-068] [SWE-104.f]
 4. Address and track defects to closure, and [SWE-030] [SWE-069]
 5. Specify where results and evaluations are retained (e.g., a test log, defect tracking system, or matrix). [SWE-065.c]
- b. Tests cases. Test cases generally include a test case identifier, the requirements addressed by the test case, prerequisite conditions, test input, expected test results, criteria for evaluating results, and test procedure/operator actions. Test case order may also need to be specified. [SWE-028] [SWE-072] [SWE-065.b] [SWE-104.h] [SWE-114.b]
 1. For each test case identify the requirement(s) that the test case verifies and document the bidirectional traceability between the test cases and the Software Requirements Specification (e.g., this can be done by constructing a traceability matrix between test cases and requirements). Multiple test cases may be needed to verify a single requirement. [SWE-028] [SWE-065.b] [SWE-072] [SWE-104.j] [SWE-114.b.2] [SWE-114.c]
 2. For each test case describe the actions necessary to verify that the software fulfills the requirement(s). Actions may include establishing initial conditions, entering inputs/issuing commands, instructions for conducting the test case, and evaluating the output against expected results and criteria. [SWE-028] [SWE-114.b] [SWE-072]
- c. Test milestones (e.g., the start and end dates for completing execution of the test plan; or the project may document these milestones in the SMP schedule). [SWE-028] [SWE 104.i]
- d. *Testing environment(s), site(s), personnel, and participating organizations. [SWE-104.k] [SWE-028] This section is completed when the project needs to plan in advance for a particular software test environment (e.g., the environment does not exist and it must be developed, environments needing the scheduling of particular resources, or specific configuration set up in order to run the tests). The testing environment includes software, hardware, and facilities necessary to run the tests, etc. Document the test resources and schedule for utilization of those resources (e.g., a lab or simulator or facility). Test preparation and configuration of the environment necessary to run the test can also be included here. [SWE-114.d]

Appendix A4: Software Requirements Specification

Instructions: The following defines the required content, but the format and style are left to the project.

The Software Requirements Specification shall contain: [SWE-049] [SWE-109]

- a. **System overview.** [SWE-109.a] Briefly state the purpose of the system and the software to which this document applies. Identify the project sponsor, acquirer, user (e.g., Principal Investigator and science team), developer/maintainer, and other stakeholders; identify current and planned operating sites; and list relevant documents. [J-STD-016:F.2.4(1)]
- b. **Functional requirements and data requirements** (e.g., this includes the functions the software must perform and the required inputs, processing, and outputs). [SWE-109.b.1] A “function” is a group of related requirements. Include those characteristics of the software item that are conditions for software item acceptance, and defer to the Software Design Descriptions those characteristics that the customer is willing to leave up to the developer. Section d. of this Software Requirements Specification provides a list of topics to be considered when specifying requirements regarding input the software item is required to accept and output it is required to produce. [J-STD-016:F.2.4(3.2)]
- c. ***Required states and modes.** [SWE-109.b.2] Examples of states and modes include: startup, calibration, housekeeping, idle, ready, active, post-use analysis, training, degraded, backup. The distinction between states and modes is arbitrary. A software item may be described in terms of states only, modes only, states within modes, modes within states, or any other scheme that is useful. If states and/or modes are required, each requirement or group of requirements in this specification should be correlated to the states and modes. The correlation may be indicated by a table or by annotation of the requirements. [J-STD-016:F.2.4(3.1)]
- d. ***External interface requirements.** [SWE-109b.3]
 1. Identify the interfacing entities (systems, hardware items, software items, users, etc.) and interfaces to which this document applies. One or more interface diagrams may be included to graphically depict the interfaces. [J-STD-016:F.2.3(3.1)] Conventions needed to understand the diagrams should be presented or referenced. If the interface operates differently in different states or modes, each requirement or group of requirements for the interface should be correlated to the states and modes. [J-STD-016:F.2.3(3.3.1)]
 2. For each interface consider defining the following:
 - (a) Priority that the interfacing entity(ies) is required to assign the interface [J-STD-016:F.2.3(3.x.a)]
 - (b) Requirements on the type of interface (such as real-time data transfer, storage-and retrieval of data, etc.) to be implemented. [J-STD-016:F.2.3(3.x.b)]
 - (c) Required characteristics of individual data elements that the interfacing entity(ies) will provide, store, send, access, receive, etc. [J-STD-016:F.2.3(3.x.c)] Consider the following when defining the requirements for the data element:
 - (1) Names/identifiers and description of content.
 - (2) Data type (alphanumeric, integer, etc.).
 - (3) Size and format (such as length, bit-level description of data interface, and punctuation of a character string).
 - (4) Units of measurement (such as meters, nanoseconds).
 - (5) Range or enumeration of possible values (such as 0-99).
 - (6) Precision (number of significant digits).
 - (7) Priority, timing, frequency, volume, sequencing, and other constraints, such as whether the data element may be updated.
 - (8) Sources (setting/sending entities) and recipients (using/receiving entities). [J-STD-016:F.2.3(3.x.c)]
 - (d) Required characteristics of data element assemblies (records, messages, files, arrays, reports, etc.) that the interfacing entity(ies) will provide, store, send, access, receive, etc. [J-STD-016:F.2.3(3.x.d)] Consider the following when defining the requirements for the data element assembly:
 - (1) Names/identifiers.
 - (2) Data elements in the assembly and their structure (number, order, grouping).
 - (3) Relationships among assemblies, such as sorting/access characteristics.
 - (4) Priority, timing, frequency, volume, sequencing, and other constraints, such as whether the assembly may be updated.
 - (5) Sources (setting/sending entities) and recipients (using/receiving entities). [J-STD-016:G.2.2(3.x.d.1-8)]
 - (e) Identify the communication methods (e.g., Ethernet, 1553 Bus, S-Band Radio) that the interfacing entity(ies) is required to use for the interface. [J-STD-016:G.2.2(3.x.e)] The project may find value in including the following items:
 - (1) Communication links/bands/frequencies/media and their characteristics.
 - (2) Message formatting.
 - (3) Flow control (such as sequence numbering and buffer allocation).
 - (4) Data transfer rate, whether periodic/aperiodic, and interval between transfers.

- (5) Routing, addressing, and naming conventions. [J-STD-016:G.2.2(3..x.e.1-8)]
- (f) Required characteristics of protocols the interfacing entity(ies) will use for the interface (e.g., Transmission Control Protocol/ Internet Protocol (TCP/IP), User Datagram Protocol (UDP)). [J-STD-016:F.2.3(3.x.f)] The project may find value in including the following:
 - (1) Priority/layer of the protocol.
 - (2) Packeting, including fragmentation and reassembly, routing, and addressing.
 - (3) Legality checks, error control, and recovery procedures.
 - (4) Synchronization, including connection establishment, maintenance, termination.
 - (5) Status, identification, and any other reporting features. [J-STD-016:G.2.2(3.x.f.1-6)]
- (g) Interface compatibility. This could be the identification of the transition mechanism/compatibility mode (e.g., hybrid dual Internet Protocol (IP) stack implementation to allow compatible communications between interfaces using IP Version 4 and IP version 6) or the identification of the subsets of functionality (to ensure compatibility between interfacing entities) in cases where both sides of the interface are using different versions of a standardized communication method/protocol. Other specifications, such as physical compatibility of the interfacing entity(ies) (dimensions, tolerances, loads, voltages, plug compatibility, etc.) may be included. [J-STD-016:G.2.2(3.x.g)]
- e. ***Adaptation requirements (data used to adapt a program to a given installation site or to given conditions in its operational environment).** [SWE-109.b.6]
- f. ***Performance and timing requirements.** [SWE-109.b.8] Specify required behavior of the software item and include applicable parameters, such as response times, throughput times, other timing constraints, sequencing, accuracy, capacities (how much/how many), priorities, continuous operation requirements, and allowable deviations based on operating conditions. Include, as applicable, required behavior under unexpected, unallowed, or "out of bounds" conditions, and requirements for error handling. [J-STD-016:F.2.4.(3.2.x)]
- g. ***Security and privacy requirements.** [SWE-109.b.9]
- h. ***Environment requirements (e.g., computer hardware and operating system).** [SWE-109.b.10]
- i. ***Design and implementation constraints (e.g., requirements that constrain the design and construction of the system such as the use of a particular data standard, existing components, use of government-furnished property (equipment, information, or software), use of particular design or implementation standards, use of particular data standards, or programming language).** [SWE-109.b.13]
- j. ***Personnel-related requirements (e.g., specific requirements for each user type such as administrator and operator).** [SWE-109.b.14]
- k. ***Training-related requirements (e.g., on line help/tutorials).** [SWE-109.b.15]
- l. ***Packaging requirements (e.g., storage limitations on deliverable (on one disk), encryption, cyclic redundancy check on the deliverable, or digital signature).** [SWE-109.b.17]
- m. ***Testing requirements that drive software design decisions (e.g., checkpoint restart, the ability to see and modify variable values, monitoring execution time or bandwidth, the ability to inject defects).** [SWE-109.f]
- n. Bidirectional traceability between this document's requirements and any higher level parent requirements documents. [SWE-109.d]

Appendix A5: Software Design Description

The Software Design Description shall include a design showing the following: [SWE-057][SWE-111]

- a. **The decomposition into software items.** [SWE_111.c] A software item is an aggregation of software, such as a computer program or database, that satisfies an end use function and is designated for purposes of specification, 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. [J-STD-016:3.1.37]
- b. **The interrelationship between software items.** [SWE_111.c] Show the static (such as “consists of”) relationship(s) of the software items. Multiple relationships may be presented, depending on the selected software design methodology (for example, in an object-oriented design, this section may present the class and object structures as well as the module and process architectures of the software item). [J-STD-016:G.2.4(4.1.b0)]
- c. **Concept of execution** (e.g., a description or diagram that explains how the software items will interact during operation, initial conditions, response to each input, sequence of operations). [SWE_111.c.1.d] [J-STD-016:G.2.4(5.x.f.4)] This could include as applicable, flow of execution control, data flow, dynamically controlled sequencing, state transition diagrams, timing diagrams, priorities among software items, handling of interrupts, timing/sequencing relationships, exception and error handling, concurrent execution, dynamic allocation/deallocation, dynamic creation/deletion of objects, processes, tasks, and other aspects of dynamic behavior. [J-STD-016:G.2.4(4.20)]
- d. **External interfaces, including Input/Output description.** [SWE_111.c.1.b] [SWE_111.c.3] External entities include items such as systems, hardware items, software items, and users. If a software item contains, receives, or outputs data, include a description of its inputs, outputs, and other data elements and data element assemblies, as applicable. [J-STD-016:G.2.4(5)] See Appendix A4.d for more detailed guidance on defining interfaces descriptions.
- e. Traceability between the Software Requirements Specification and the Software Design Description.
- f. Include the software design decisions (e.g., assumptions, limitations, and reliability related items/concerns or constraints) and rationale for those decisions. [SWE_111.a] [SWE_111.c.2]

Note: For the more complex projects, a more detailed design description with embedded guidance is provided in LMS-CP-7150.4: Class C Software, Appendix A7.

Appendix A6: Software Version Description

The Software Version Description shall contain: [SWE-063] [SWE_116]

- a. Software name and the version identifier to which this Software Version Description applies (e.g., Software X – Version 9.2, Software X-MM/DD/YY, Software X- Release 2). [SWE_116.a] If there are multiple systems within the target environment(s) or the software configuration item is customized for a particular hardware configuration, determine if the system identification also needs to be included.
- b. ***Summary of updates/changes since the previous Software Version Description, any open defects, and workarounds.** [SWE_116.d] [SWE_116.h] [SWE_116.i] This could be a high-level summary of the enhancements and fixes or the project's current list of changes and defects and their status.
- c. **Instructions for building the executable software**, including, for example, the instructions and data for compiling and linking and the procedures used for software recovery, software regeneration, testing, or modification. [SWE_116.e]
- d. ***If the Software Version Description is not co-located with the software product files, include the location of the files.** [SWE_116.g] Software product files are the set of computer programs, procedures, and associated documentation and data. Examples include requirements, design, source code, object code, databases, test information, batch files, command files, data files, manuals, and any files needed to install, build, operate, and maintain the software. [SWE_116.b] [SWE_116.g]

Appendix B: Plan Exemptions

a. Routine maintenance exemptions.

Routine maintenance is 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] The following Appendix A1, Software Management Plan, items and subitems are optional for routine maintenance only projects:

- d. Engineering environment,
- e.1. Document a list (e.g., Work Breakdown Structure (WBS)),
- e.2.a. Coordinates with the overall project schedule,
- e.2.c. Documents milestones and delivery dates,
- l. Software development life-cycle phases.

Appendix C: LaRC Compliance Matrix for Class D Software (not Safety Critical)

Instructions: For each STEP and Appendix listed below, complete the 3rd and 4th column of this matrix or equivalent; complete all remaining columns for each tailoring request. Obtain the approvals listed at the bottom of the matrix. See "Appendices for LMS-CP-7150.5: Class D Software" at: <https://sites-e.larc.nasa.gov/sweng/supporting-products/> for a Microsoft Word electronic copy of this Matrix.

Note: To add additional rows within a STEP, right click on a cell in the middle of the row, select "Insert," select "Insert Rows Above" or "Insert Rows Below"; follow a similar process for adding columns.

Name of Project: _____

Date Approval Requested: _____

[SWE-125]

LMS Procedure		Planned Implementation		Tailoring		
Step #	Step Name	Step or substep ID#	Responsible party	If an LMS CP step is tailored, explain the tailoring requested	Impacts/risks associated with the tailoring request	Justification for tailoring requests (why impacts and risks are acceptable)
1	Assess options for software development vs. acquisition					
2	Acquisition preparation					
3	Develop software plans					
4	Review Software Management Plan (SMP) and Compliance Matrix					
5	Execute plans and manage software activities					
6	Ensure requirements and test documentation is maintained					
7	Develop Software Requirements Specification					
8	Develop Software Design Description					
9	Develop Test Plans concurrently with Steps 7, 8, & 10					
10	Develop, verify, & validate code					
11	Deliver Software Products or Data					

LMS Procedure		Planned Implementation		Tailoring		
Appendix #	Appendix name	Appendix item or subitem ID#	Responsible party	If an LMS CP Appendix item is tailored or Not Applicable, explain tailoring request or mark it as NA ¹	Impacts/risks associated with the tailoring request	Justification for tailoring requests (why impacts and risks are acceptable)
A1	Software Management Plan (SMP)					
A2	Software Configuration Management Plan					
A3	Software Test Plan					
A4	Software Requirements Specification					
A5	Software Design Description					
A6	Software Version Description					

Approvals Required for Planned Implementation

²Software Manager: _____ Date: _____ Approved (Yes, No) Sig.³ _____
⁴Software Manager's Line Manager: _____ Date: _____ Approved (Yes, No) Sig.³ _____

Additional Approvals Required if Tailoring is Requested (Follow requirements for obtaining approvals in *LPR 7150.2, Section 2. Tailoring and Waivers.*)

⁵Applicable project personnel: _____ Date: _____ Approved (Yes, No) Sig.³ _____
 Mission Assurance Branch (C201): _____ Date: _____ Approved (Yes, No) Sig.³ _____
⁶Software Engineering Process Group Rep: _____ Date: _____ Concurred (Yes, No) Sig.³ _____
⁷Technical Authority (Directorate Head): _____ Date: _____ Approved (Yes, No) Sig.³ _____
⁸Other: _____ Date: _____ Approved (Yes, No) Sig.³ _____

¹ As specified in Appendix A of LMS-CP-7150.5: Class D Software, only those requirements in Appendix A1 thru A6 marked with an asterisk "*" may be denoted as "NA."
² Approval by the Software Manager confirms that the project plans to complete all LMS-CP-7150.5 requirements and any requested tailoring specified in the above Compliance Matrix.
³ Optional: Written or electronic signature.
⁴ The Line Manager reviews & approves the SMP and Compliance Matrix to ensure the project complies with LMS-CP-7150.5, Class D Software, and to approve tailoring requests.
⁵ Individuals accepting the risk associated with the tailoring.
⁶ The Software Engineering Process Group representative from the software manager's Directorate.
⁷ The Software Manager's Directorate Head.
⁸ This may be the LaRC Director SMA Office, NASA HQ CE, or HQ Chief SMA (see LPR 7150.2A for approvals required).