

Table of Contents

Section/Title
SOFT-GEN: General Software Requirements (READ FIRST)
Appendix A: Chapter 21 Definitions and Acronyms
Appendix B: SO and SRLM Decision Diagram for FAC-COE
Appendix C: Chapter 21 References
Attachment 1: Chapter 21 Summary Table
SOFT-GEN-FM01: Software Baseline Form SOFT-GEN-FM01(I): Instructions with Example
SOFT-GEN-FM02: Non-SSC Software Change Package Form SOFT-GEN-FM02(I): Instructions
SOFT-GEN-FM03: Software Data Sheet Form SOFT-GEN-FM03(I): Instructions with Example
SOFT-ACQUIRE: Acquisition
SOFT-DESIGN: Design
SOFT-V&V: Verification & Validation
SOFT-V&V-FM01: Software Approval for Use Form SOFT-V&V-FM01(I): Instructions With Example
SOFT-INV: Inventory
SOFT-MAINT: Use, Maintain & Retire

*This search tool may not include the latest files; use individual files for work.*

**TABLE OF CONTENTS**

1.0 PURPOSE, SCOPE, APPLICABILITY AND ORGANIZATION..... 2

1.1 Chapter 21 Purpose, Scope, and Applicability ..... 2

1.2 Chapter 21 Organization and Use..... 3

1.3 Section SOFT-GEN Purpose, Scope, and Applicability..... 6

2.0 SOFTWARE IDENTIFICATION AND DETERMINATION..... 9

2.1 Software Identification ..... 9

2.2 Software Determination ..... 10

3.0 SOFTWARE DATA SHEET (SWDS) ..... 11

4.0 CONFIGURATION MANAGEMENT (CM)..... 12

4.1. Develop and Maintain a Software Baseline (SWBL) ..... 12

4.2. Control Software Changes..... 14

5.0 PERFORM PROBLEM REPORTING & CORRECTIVE ACTION..... 16

6.0 PERFORM SOFTWARE (PROJECT) RISK MANAGEMENT ..... 17

7.0 APPENDICES, ATTACHMENTS AND FORMS..... 18

APPENDIX A: CHAPTER 21 DEFINITIONS AND ACRONYMS ..... 19

APPENDIX B: SO AND SRLM DECISION DIAGRAM FOR FAC-COE ..... 27

APPENDIX C: CHAPTER 21 REFERENCE LISTING ..... 29

ATTACHMENTS:

ATTACHMENT 1: CHAPTER 21 SUMMARY TABLE

FORMS:

- SOFT-GEN-FM01: SOFTWARE BASELINE FORM (SWBL)
- SOFT-GEN-FM02: NON-SSC SOFTWARE CHANGE PACKAGE FORM (SWNCP)
- SOFT-GEN-FM03: SOFTWARE DATA SHEET FORM (SWDS)

Rev	Date	Description	POC	RM
0	06/23/16	Initial issue	Tobin Oruch, ES-DO	Mel Burnett, CENG-OFF

**Note:** This entire chapter is currently a provisionally issued document. In this case, that means chapter use is optional until SOFT-GEN is revised to remove provisional markings (expected in approximately 4-8 months from date of Rev. 0 of this section). Managers may invoke chapter sooner (also, training for software managers and owners is being developed). In other words, assuming no intermediate provisional revisions of SOFT-GEN, Rev. 1 of SOFT-GEN will not have this note and, thus, the entire chapter will become mandatory.

As with all LANL Engineering Standards, but especially with this all-new chapter, please contact the Chapter [POC](#) with comments, issues, etc.

## 1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16  
PROVISIONAL CHAPTER

## 1.0 PURPOSE, SCOPE, APPLICABILITY AND ORGANIZATION

## 1.1 Chapter 21 Purpose, Scope, and Applicability

A. Chapter 21 is the software plan for the Facility Conduct of Engineering program (FAC-COE)<sup>1</sup>, including safety basis activities. It provides reasonable assurance that in-scope software will consistently, compliantly, and efficiently satisfy its intended use. It describes the “how, when, and who” for implementing the software requirements of:

1. DOE Order [414.1D](#), *Quality Assurance*, (hereafter O 414.1D) and
2. [ASME NQA-1-2008/NQA-1A-2009](#), *Quality Assurance Requirements for Nuclear Facility Applications, Part I and Part II* (hereafter NQA-1<sup>2</sup>).

B. For new and modified software, this chapter applies only when software is either:

1. Within the scope of [PD340](#), *Conduct of Engineering for Facility Work*, or
2. Within the scope of [PD110](#), *Safety Basis*.

Note: Modified software includes Major computer program changes (see Definitions). Both major computer program changes (including upgrades from non-safety to safety software) and software plans considered non-compliant<sup>3</sup> are subject to this chapter. LANL: Once the need to adopt this chapter for a given software is identified, initiate a PFITS action to resolve deficiencies versus the chapter with a due date agreed to by both SRLM and Chapter POC.<sup>4</sup>

C. Exclusions: This chapter does not apply to:

1. Software covered by existing, compliant software quality management plans/processes. The issuance of this chapter does not necessitate immediate replacement of those controls (grandfathering applies) unless/until a Major change;
2. Programmatic R&D engineering that does not affect the safety basis or technical baseline configuration of a LANL facility (nor is described in the Hazard Analysis (HA) or [is] adjacent [to nuclear safety] structures, systems, and components (SSCs)<sup>5</sup>. (Such work is conducted pursuant to the provisions of [PD370](#), *Conduct of Engineering for Research and Development [R&D]*);
3. Uncomplicated software tools, including productivity aids that are adequately controlled through the design process. Examples include unit conversion software, spike computer programs, equipment selection software, diagnostic software, or interpolation calculators; and/or,
4. Personal productivity and collaboration software that does not provide calculation output.

<sup>1</sup> FAC-COE defined by [PD340](#), *Conduct of Engineering for Facility Work*, and clarified by ES-DO: 15-006 (EMRef-74) to also include the first parenthetical in 1.1.C.2.

<sup>2</sup> [NQA-1](#) requirements are clarified for LANL application in [SD330](#), *LANL Quality Assurance Program*, and [P1040](#), *Software Quality Management (SQM)*.

<sup>3</sup> Compliant means meeting applicable DOE O 414.1D and NQA-1 SQA requirements. Compliance with LANL P1040 is also acceptable. When in doubt about compliance of a given plan, assess against relevant requirements (QPA may be able to assist).

<sup>4</sup> Six months is believed appropriate for most software while some may justify longer.

<sup>5</sup> Criteria from PD340 and ES-DO: 15-006. Excluded unless electively invoked by the Responsible Line Manager (RLM).

## 1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16  
PROVISIONAL CHAPTER

- D. The chapter defines and applies to two major types of software:
1. System, structure, and component (SSC) software; and,
  2. Non-SSC software (see definitions), including “simple and easily understood” software used in the design of SSCs.<sup>6</sup>
- E. Subcontractors (architect-engineers, consultants, etc.): For software within the applicability statements above, including both SSC and Non-SSC software, Chapter 21 provides minimum expectations for software quality assurance (SQA).
1. For software used for safety design or analysis, it includes program expectations and basic reporting requirements to facilitate oversight.
  2. For SSCs with software being specified or designed, and for any software being delivered for LANL ownership and use, Chapter 21 requires the same documentation deliverables that LANL personnel provide to the extent the information is known in the design and construction phases.<sup>7</sup> When not known, insert TBDs and deliver as “Draft.”
  3. Subcontractor quality assurance programs must meet the expectations of this chapter, comply with [O 414.1D](#), and where required by subcontract -- [NQA-1](#), including Part I and Part II, Subparts 2.7 and 2.14.

## 1.2 Chapter 21 Organization and Use

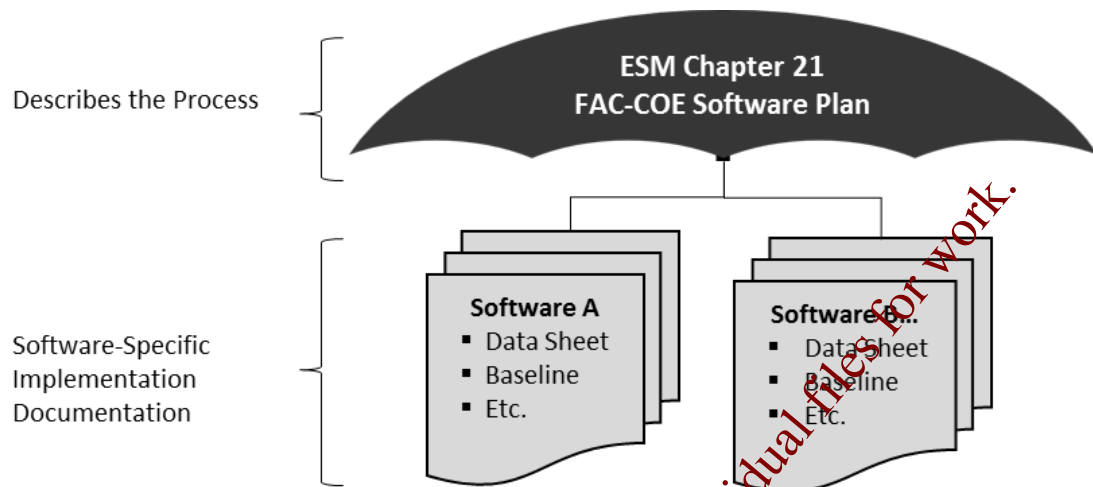
- A. Concept: As shown in Figure 21.1-1, *Chapter 21 Software Plan Concept*, this chapter uses an umbrella concept for SQA. That is, Chapter 21 serves as the software plan for software within its scope. Implementation of this chapter/plan will produce software-specific SQA implementation documentation (e.g., software baseline, software data sheet) for multiple software items. The software-specific documentation includes information that is unique<sup>8</sup> to each software program. Examples of software-specific information include the software technical requirements, software ownership, and software test plans.
- B. Organization: Chapter 21 is divided into six (6) sections that span the software lifecycle. These sections and a summary of deliverables are depicted in Figure 21.1-2, *Chapter 21 Sections and Deliverables*.<sup>9</sup>

<sup>6</sup> See definitions in SOFT-GEN Appendix A, *Chapter 21 Definitions and Acronyms*. Includes safety and non-safety software. Also, see applicability of quality assurance controls summarized in SOFT-GEN Attachment 1, *Chapter 21 Summary Table* and further details in Chapter 21 sections.

<sup>7</sup> For SSC software: Use of chapter forms/doc numbers provides standardization and retrievability of documents that LANL must maintain going forward. As necessary, documentation will be updated or produced by construction subcontractors and verified by LANL in the start-up/commissioning process in conjunction with the system engineer function.

<sup>8</sup> Unique in that it cannot otherwise readily be addressed on a higher level (in this chapter proper). This centralized+distributed model ensures consistency of approach/rigor within user organizations while eliminating the production of standalone SQA plans that repeat boilerplate; it also utilizes a living, stand-alone documentation approach for those documents that are subject to change, easing revision of same without necessitating revision of an entire plan. Also, revision of Chapter 21 does not invalidate the approach or documentation associated with a previous chapter revision unless the revised chapter material specifically states this.

<sup>9</sup> The 6 phases encompass all life cycle phases described by other industry documents in a simplified approach.

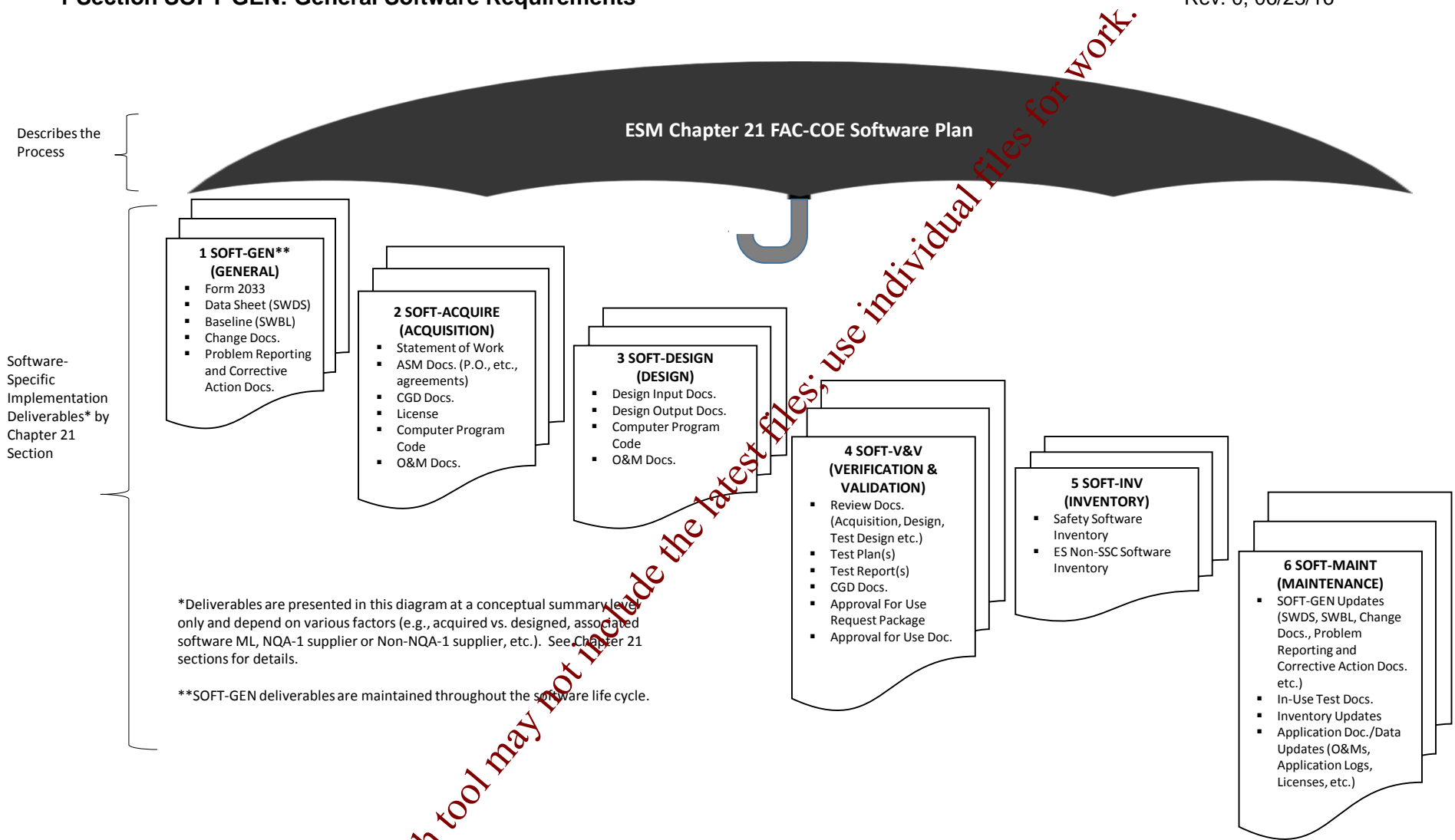


**Figure 21.1-1 Chapter 21 Software Plan Concept**

- C. Refer to SOFT-GEN, SOFT-V&V, SOFT-INV, and SOFT-MAINT for all software within the chapter scope. Also follow SOFT-Acquire for software that is acquired, and SOFT-DESIGN for software that is designed (term used interchangeable with developed).
- D. Definitions and Acronyms: See SOFT-GEN Appendix A, *Chapter 21 Definitions and Acronyms* for key definitions and acronyms. Defined terms are often capitalized, set in quotes, or both in this chapter.
- E. Refer to SOFT-GEN Attachment 1, *Chapter 21 Summary Table*, for a summary of this entire chapter. Refer to the table(s) at the beginning of each section for a summary of that section. These tables address both SSC software and Non-SSC software.
- F. Italics: Throughout the ESM, italics are one method used to indicate guidance. Exception: When italics are used to highlight document titles.
- G. Graded Approach: Management levels (MLs) -- rather than software type, category, or Software Risk Level (SRL) terms -- are used throughout.<sup>10</sup> See Form 2033 for an ML crosswalk (guidance table) to software types, categories and software risk levels (SRLs). This may necessitate performance of an ML determination using AP-341-502 when the software may be a lower ML level than the system it supports.<sup>11</sup>
- H. Roles, Responsibilities, Authorities, and Accountabilities: See SOFT-GEN Appendix B, *SO and SRLM Decision Diagram for FAC-COE*, for guidance on determining these key positions.

<sup>10</sup> Promotes a common vernacular for users.

<sup>11</sup> Current AP-341-502 excludes software but revision is planned; proceed with determination in the meantime.



\*Deliverables are presented in this diagram at a conceptual summary level only and depend on various factors (e.g., acquired vs. designed, associated software ML, NQA-1 supplier or Non-NQA-1 supplier, etc.). See Chapter 21 sections for details.

\*\*SOFT-GEN deliverables are maintained throughout the software life cycle.

Figure 21.1-2 Chapter 21 Sections and Deliverables

## 1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

- I. Or Equivalent: When this term is used, the Software Responsible Line Manager (SRLM) determines equivalency. If questions arise regarding equivalency, the Chapter 21 Point of Contact (POC) will resolve.
- J. References: See SOFT-GEN, Appendix C, *Chapter 21 Reference Listing* for hyperlinked references. Additional hyperlinks are provided in key locations throughout the chapter for convenience.
- K. Forms:
  - 1. LANL personnel: Endeavor to use Chapter forms as-is and report issues and improvement ideas to the Chapter 21 POC.<sup>12</sup> POC may authorize other methods equivalent to chapter forms in writing.
  - 2. LANL subcontractors must use Chapter forms to satisfy Chapter requirements for SSC software. For Non-SSC software, subs may either use their own forms or integrate, adapt, and reformat the forms; either approach is acceptable so long as key functions, data, and approvals are retained.
  - 3. LANL Form 2033 must be used as-is.
- L. Upkeep: So long as NQA-1 compliance is maintained, the POC is authorized to resolve conflicts within the Chapter in a manner that may not be most stringent; similarly, the Standards Manager is authorized to approve and issue revised forms, references, and GEN Att. 1 summary independently of associated Sections.

**1.3 Section SOFT-GEN Purpose, Scope, and Applicability**

- A. SOFT-GEN provides general software information and requirements to implement Chapter 21. These general requirements include how to: (a) identify software and make software determinations (e.g., Form [2033](#)), (b) implement configuration management (e.g., baselines and software changes), (c) report problems/take corrective actions, and (d) perform software risk management.
- B. See Table 21.1-1 for a summary of SOFT-GEN. For SOFT-GEN applicability, see the Chapter 21 applicability subsection above and the following:
  - 1. Other than for software identification and determination, SOFT-GEN does not apply to “Simple and Easily Understood” software used in the design of SSCs if the software is individually verified as described in SOFT-V&V.
  - 2. SOFT-GEN applies to Subcontractors providing design agency or analysis services to LANL<sup>13</sup>.

<sup>12</sup> Forms are designated as samples which means minor aspects may be modified so long as all functions, key data, and approvals are retained. Alternatively, LANL user organizations may direct that other, comparable forms be used (with prior permission of the Chapter POC). Forms may be revised administratively and reposted without revision of associated section (and with only POC and Standards Manager approval).

<sup>13</sup> SSC and Non-SSC software resulting from and/or used in the SSC design must be identified and software determinations completed as stated herein.

1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

Table 21.1-1 Section SOFT-GEN Summary for SSC Software and Non-SSC Software (This table is a summary only and does not include all requirement details. See text for details.)										
Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			Reference	
		1	2	3	4	How	When	Who <sup>2,3</sup>	P1040 R9 Ref. Section	ESM Ch. 21 Ref. <sup>4</sup>
1	Identify software and complete software determinations	R	R	R	G	<ul style="list-style-type: none"> <li>Identify SO and SRLM</li> <li>Develop software list</li> <li>Determine reasonable probability</li> <li>Complete and retain Form <a href="#">2033</a></li> </ul>	<ul style="list-style-type: none"> <li>Earliest practical time</li> <li>Before software design, acquisition</li> </ul>	<ul style="list-style-type: none"> <li>SO (D)</li> <li>SRLM (R, A)</li> <li>FDAR (R, A)</li> </ul>	3.2	SOFT-GEN, 2
2	Prepare Software Data Sheet (SWDS)	R	R	R	-	<ul style="list-style-type: none"> <li>Develop Software Data Sheet (SWDS), SOFT-GEN-FM03</li> </ul>	<ul style="list-style-type: none"> <li>Earliest practical time</li> <li>Before design, acquisition</li> <li>Maintain throughout software lifecycle</li> </ul>	<ul style="list-style-type: none"> <li>SO (D)</li> <li>SRLM (R, A)</li> </ul>	3.3.3	SOFT-GEN, 3.0
3	Develop and maintain software baseline	R	R	R	-	<ul style="list-style-type: none"> <li>Develop software baseline (SWBL) using SOFT-GEN-FM01</li> <li>Maintain baseline accuracy</li> <li>Document baseline information in the computer program code (where possible)</li> </ul>	<ul style="list-style-type: none"> <li>Earliest practical time</li> <li>Before/after acceptance testing</li> <li>Before software approval for use (SWAU)</li> <li>After approved changes</li> <li>For ML-1/ML-2 designed software, at major design activity</li> </ul>	<ul style="list-style-type: none"> <li>SO (D)</li> <li>SRLM (R, A)</li> </ul>	3.3.3	SOFT-GEN, 4.0

This search tool may not include the latest files; use individual files for work.



1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

Table 21.1-1 Section SOFT-GEN Summary for SSC Software and Non-SSC Software (This table is a summary only and does not include all requirement details. See text for details.)										
Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			Reference	
		1	2	3	4	How	When	Who <sup>2,3</sup>	P1040 R9 Ref. Section	ESM Ch. 21 Ref. <sup>4</sup>
4	Control software changes (change control)	R	R	R	G	<ul style="list-style-type: none"> <li>Document changes in computer program code (where possible)</li> <li>For SSC software, prepare changes per Table 21.1-4;</li> <li>For Non-SSC software, prepare SWNCP per SOFT-GEN-FM02</li> <li>Make doc. changes per governing document control process</li> <li>Revise software baseline</li> </ul>	<ul style="list-style-type: none"> <li>In a timely manner</li> <li>When required to ensure proper performance</li> <li>Per Table 21.1-4</li> </ul>	<ul style="list-style-type: none"> <li>Per Table 21.1-4 or equivalent</li> </ul>	3.3.3	SOFT-GEN, 4
5	Perform problem reporting and corrective action	R	R	R	G	<ul style="list-style-type: none"> <li>Before approval for use (SWAU), use <a href="#">P330-6</a>, <a href="#">P332-4</a>, less formal methods (e.g., bug lists, comments in program) Or Equivalent</li> <li>After SWAU, use <a href="#">P330-6</a>, <a href="#">P322-4</a> Or Equivalent</li> </ul>	<ul style="list-style-type: none"> <li>Throughout software lifecycle</li> </ul>	<ul style="list-style-type: none"> <li>SO (D)</li> <li>SRLM (R, A)</li> </ul>	3.3.3	SOFT-GEN, 5
6	Perform software risk management	R	R	R	G	<ul style="list-style-type: none"> <li>Implement Chapter 21</li> <li>Develop and maintain risk list as part of SWDS</li> </ul>	<ul style="list-style-type: none"> <li>Throughout software lifecycle</li> </ul>	<ul style="list-style-type: none"> <li>SO (D)</li> <li>SRLM (R, A)</li> </ul>	3.3.3	SOFT-GEN, 6

Notes:

<sup>1</sup> ML = Associated Management Level as per Form 2033. R = Required. G = Required but graded. "-" = Not required.

<sup>2</sup> D = Develop; R = Review; A = Approve.

<sup>3</sup> SO = Software Owner. SRLM = Software Responsible Line Manager. FDAR = Facility Design Authority Representative.

<sup>4</sup> Ref. = ESM Chapter 21 section name and subsection number (e.g., Section SOFT-GEN, Subsection 3, Software Identification and Determination).

This search tool may not include the latest files; use individual files for work.

## 2.0 SOFTWARE IDENTIFICATION AND DETERMINATION

### 2.1 Software Identification

#### A. How

1. For Non-SSC ML-1 through ML-4 software that is planned to be acquired or designed, identify and summarize the software.<sup>14</sup>

Develop a list of the software. Include the following on the list: (a) software name, (b) software functional description (what it does), (c) software application (where/how used), (d) whether the computer program can be changed other than through replacement of the software (replacement includes software upgrades).

2. For SSC ML-1 through ML-3 software that is planned to be acquired or designed, identify and summarize software as described above with the following clarification (Note: For ML-4 SSCs, this chapter does not require software identification; however, it may be invoked by the SRLM):

- a. Analyze the SSC to determine whether the SSC has software. Visually inspect and/or review vendor information (product specifications, catalog data, operations and maintenance manuals, etc.). Identify to a practical extent and to a level that the software can be controlled. *For example, for a control system module that has multiple pieces of software on a single module and the software cannot be modified other than through replacement of the entire module, then identifying the software to the module part number is appropriate.* Do not introduce a potential for: adversely affecting manufacturer warranties, SSC damage, or unintended software changes.

3. Provide the list to the LANL Software Responsible Line Manager (SRLM). See Appendix B for guidance in determining the SRLM.<sup>15</sup>

B. When. For Non-SSC software, provide the software list to the LANL SRLM prior to software acquisition or design. For SSC software, provide draft lists with 60% and final with 90% detail design deliverables (Ref. ESM [Chapter 1](#), Section Z10, *General*). For deferred SSC design, include in deferred design submittals.

C. Who. The Software Owner (SO) is responsible for software identification. See Appendix B for guidance on determining the SO.

<sup>14</sup> The software list is not a controlled document or quality record. The responsible LANL SRLM uses this information to ensure that software determinations and subsequent required actions in this chapter are completed.

<sup>15</sup> For Subcontractors, listing to be provided per 30-60-90% deliverable schedule; LANL project engineer function will ensure review by appropriate SMEs and/or FDAR.

2.2 Software Determination

A. How. Use Table 21.1-2, and the following text, to complete software determinations.

1. Ask the following question as shown in Table 21.1-2:

Table 21.1-2 The Reasonable Probability Software Question
<p>Excluding personal productivity and collaboration software that does not provide calculation output (e.g., e-mail software, presentation software)<sup>1</sup>, is or will the software be used in connection with<sup>3</sup> the design, analysis, and/or operation of:</p> <ul style="list-style-type: none"> <li>▪ a nuclear (including radiological) facility (Ref. <a href="#">LANL Nuclear Facility List</a> and <a href="#">Conduct Of Operations Resources</a> websites);</li> <li>▪ an accelerator, live-firing range, biological hazard facility, high explosive facility, or moderate- or high- chemical hazard facility<sup>2</sup> as determined using Safety Basis Procedure (<a href="#">SBP111-1</a>, Facility Hazard Categorization and Documentation); or</li> <li>▪ LANL's Essential Functions (EFs) as described in <a href="#">EPP-COOP-006</a>, <i>LANL COOP [Continuity of Operations] Plan</i> ?</li> </ul>
<p>Notes:</p> <p><sup>1</sup>Personal productivity and collaboration software that does not provide calculation output is non-safety, commercially controlled software.</p> <p><sup>2</sup> From <a href="#">SBP111-1</a>, a facility is defined as equipment, structure, system, process, or activity that fulfills a specific purpose except for utility distribution structures.</p> <p><sup>3</sup>Use must be associated with ML-1, ML-2 or ML-3 items. The term “in connection with” captures software that directly or indirectly is used in relation to design, analysis or operation. Operation includes SSC operation (e.g., valve actuation) and administrative operation (e.g., combustible load tracking software).</p>

If the answer to the question is “yes” or “not sure” and the associated SSC ML is ML-1 through ML-3 or unknown, then complete Form [2033](#). If the associated SSC is ML-4, then do not complete Form [2033](#) unless required by the SRLM or Subcontract.

By default, all software is non-safety and commercially controlled (analogous to ML-4) unless there is a reasonable probability that the software could be safety software or risk significant software.

*Guidance: Examples of commercially controlled, non-safety software include: Microsoft PowerPoint® presentation graphics program, Microsoft Outlook® messaging software.*

Complete Form [2033](#) using the instructions on the form. See form instructions for which revisions of the form may be used. If the software is planned to be used as “simple and easily understood software” used in the design of SSCs (see definitions), then note it on the form.

*Guidance: Form 2033 allows multiple software items to be included (often it is the support software), and SOFT-INV allows SSC software to be inventoried as a software collection (bundled); see 3.0, A. 5. Bundling may work better if all software is the same ML level relative to 2033 form and subsequent Ch. 21 requirements (may cause confusion if not). Example of where bundling might be considered: A control panel that has a PLC and a*

## 1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

*touchscreen and separate programs for these components, those separate programs, both ML-2, could be given a single SWID (and 2033, etc.)*

3. Obtain a software identification number (SWID) in accordance with [AP-341-402](#), *Engineering Document Management in Operating Facilities*. Ensure the SWID is part of the Form [2033](#) record number and subsequent software documentation. Retain the completed Form [2033](#) as a record in accordance with the SRLM's governing records management process.

B. When.

1. Complete at the earliest practical time before software design or acquisition<sup>16</sup>.
2. If the application of the software significantly changes, review the original determination and complete a new Form [2033](#) as required. *Guidance: It is recommended that Form [2033](#)s are reviewed at least every three years or other time period that is commensurate with the associated risk of the software.*

**Note:** Changes made to the software (i.e., software version) do not require a new Form [2033](#) to be completed unless the change significantly affects the application. The software inventory must be reviewed and updated as required. See SOFT-INV.

C. Who. The SRLM is responsible for software determinations.

### 3.0 SOFTWARE DATA SHEET (SWDS)

- A. How. For ML-1 through ML-3 software, prepare a Software Data Sheet (SWDS). See SOFT-GEN-FM03, *Software Data Sheet Form (SWDS)*. One SWDS can be used for multiple installations. Process and retain the data sheet in accordance with the SRLM's governing document control records management processes (e.g., *EDMS*, and for *ES-Div*, *associated Desktop Instruction*).

Ensure the correct SWID (obtained when completing Form [2033](#)) is part of the SWDS record number and subsequent software documentation.

**ML-4 Note:** Software data sheets are not required for ML-4 software however may be used at the SRLM's discretion.

- B. When. Complete a SWDS at the earliest practical time and before design and/or acquisition as applicable (e.g., at end of planning).<sup>17</sup> Revise the SWDS as required to maintain accuracy.

**Note:** *It is recommended that the SRLM review the data sheet at least every two years.*

Who. At a minimum, the SO develops and the SRLM reviews and approves the SWDS.

<sup>16</sup> For SSC software, submit draft 2033 forms with the 60% detail design submittals and finals with the 90% submittal. Indicate "To be determined (TBD)" where information is not available.

<sup>17</sup> Ref. IEEE 12207. At the end of the planning phase, SWDS is proof of a plan and management commitment to the project.

#### 4.0 CONFIGURATION MANAGEMENT (CM)

- A. Maintain software configuration management (CM) in accordance with this Subsection<sup>18</sup>.

#### 4.1. Develop and Maintain a Software Baseline (SWBL)

- A. How.

1. For ML-1 through ML-3 software, develop and maintain a software baseline. ML-4 software does not require a software baseline unless it is required by the LANL SRLM. Use SOFT-GEN-FM01, *Software Baseline Form (SWBL)*. The software baseline may be a standalone document, or it may be part of other documents.
2. Integrate the software baseline, as applicable with SSC technical baseline documents. See *Engineering Standards Manual Chapter 1, Section Z10, General*; [AP-341-616](#), *Technical Baseline Change During Design*; and [AP-341-405](#), *Identification and Control of Technical Baseline, Variances, Alternate Methods, and Clarifications in Operating Facilities*.
3. As applicable for SSC software, ensure configurations (e.g., set points) are established in accordance with [AP-341-613](#), *Instrumentation Set Point Control*.
4. Document the baseline configuration items needed to run the program. Include, as applicable<sup>19</sup>:
  - a. **Documentation** - software design requirements; instructions for computer program use including installation and test requirements; system description; test plans and results; characteristics derived from regulatory requirements and commitments; calculations and analyses; support software documentation.
  - b. **Computer program(s)** - source, object, back-up, data,<sup>20</sup> configuration, and support software files needed to run the program. Configuration files include limits, settings, set points, etc.
5. Develop and control the software baseline in accordance with the SRLM's document control process (e.g., *EDMS*). Ensure only authorized changes are made to software baselines.

Ensure the correct SWID (obtained when completing Form 2033) is part of the SWBL record number and subsequent software documentation. See [AP-341-402](#) for details.
6. Update the software baseline after software changes. Ensure that the baseline defines the most recent approved software configuration.<sup>21</sup>

<sup>18</sup> See [ANSI/IEEE Std. 828](#), *IEEE Standard for Configuration Management in Systems and Software Engineering* for guidance.

<sup>19</sup> Applicability is the determination that the activities and processes are actually used. For example, if the work scope does not include or contribute to design, then design requirements do not apply and the related design procedures are not used [Ref. [SD330](#), *LANL Quality Assurance Program*].

<sup>20</sup> This is the data needed to run the program. This can include data such as limits, set points, and databases.

<sup>21</sup> In the event the in-use computer program fails, becomes corrupted, or other problems occur, the baseline may be used to re-establish operation

7. Identify the associated hardware needed to run the computer program indirectly in the system design description document(s) or directly in the baseline.
8. When “Support Software” (including software tools) is used for new or modified software, one must evaluate, review, accept for use and place it under configuration control as well. Note however, that software tools that do not affect the performance of the software need not be placed under configuration control. Identify the software/tool in the baseline and manage the configuration. Changes to software tools must be evaluated for impact on the software product to determine the level of reviews and retesting that will be required. Vendor-supplied software used with hardware to configure control systems is an example of support software.
9. Implement a baseline labeling system in accordance with the following:
  - a. Uniquely identify each configuration item (documentation or computer program).
  - b. Identify changes to configuration items by revision.
  - c. Provide the ability to uniquely identify each configuration of the revised software available for use.
  - d. For baseline document labeling, use the SRLM’s approved document numbering system (e.g., [AP-341-402](#), *Engineering Document Management in Operating Facilities*).
  - e. Include baseline label information within the computer source code where possible. For each logical block or class, include a brief description of its function, the name of the person writing the description and the date the description was added.. For code changes following SWAU, update the header comments with the revision number (e.g. 1.01, 1.02...) and the SCP number driving the change. Guidance: *Include a brief description of the block function, the name of the person providing the description and the date the description was provided.*
  - f. For off the shelf software (OTS) or other (existing) software with a compliant labeling configuration, use the OTS or existing software labeling configuration.
  - g. For other software not described in (f), use Table 21.1-3, *Computer Program File Labeling System*, or as approved by the Chapter 21 POC.
  - h. Ensure the correct SWID (obtained when completing Form 2033) is integrated into the baseline labeling where possible.

Table 21.1-3 Computer Program File Labeling System			
Prior to Approval for Use <sup>1</sup>		After Approval for Use	
Major Change <sup>2</sup>	Minor Change <sup>2</sup>	Major Change	Minor Change
A, B, C...	A.01, A.02, A.03...	1, 2, 3...	1.01, 1.02, 1.03...
<sup>1</sup> Software approval for use (SWAU) constitutes that the software requirements have been satisfied (including testing, user's manuals, etc.) and the software is ready to be used in its intended operating environment.			
<sup>2</sup> See definitions of major change and minor change in the definitions section.			

B. When.

1. For ML-1 and ML-2 designed software, establish software baselines at the completion of each major software design activity.
2. Establish software baselines as soon as practical. At a minimum, establish before formal acceptance testing/commissioning. Pre-acceptance testing may be performed to support verification of the baseline prior to (final) acceptance testing. Maintain baselines (configuration items) under configuration management until software retirement.

C. Who. The SO prepares the baseline. The SRPM and others as required in the governing document control process, review and approve the software baseline.

## 4.2. Control Software Changes

A. How.

1. Safety Software: For computer program changes (revisions) to software, use SOFT-V&V to V&V the software change and/or commercial grade dedication (CGD), e.g., per STD-341-703, to dedicate the change. Unless a more conservative approach is required by the LANL SRLM, use the following graded approach for dedicating software changes:
  - a. Dedicate Major Changes (see Definitions).
  - b. Dedicate Minor Changes (see Definitions) if the changes affect the critical characteristic(s) that pertain to the functionality of the computer program and as applicable, the associated SSC.
2. For all ML levels, see Table 21.1-4 and apply the following to Major and Minor software changes:
  - a. For designed software, institute design control measures commensurate with those applied to the original design.
  - b. Document changes and include the following in the change documentation:
    - 1) a description of the change;
    - 2) rationale for the change;
    - 3) identification of affected software baselines; and
    - 4) evaluation and approval of the change (see SOFT-V&V for evaluation methods/criteria).

Table 21.1-4 SOFT-GEN Software Change Summary by Software Management Level (ML) (This is a summary only and does not include all requirement details. See text for details.)									
For the following types of software changes	Provide the following deliverable	SSC Software ML <sup>1</sup>				Non-SSC Software ML <sup>1</sup>			
		1	2	3	4	1	2	3	4
Less than minor change, minor and major changes made <u>before or after</u> approval for use (SWAU)	Documented comment in the computer program code where feasible	R	R	R	R	R	R	R	R
Minor or major changes made during design implementation and <u>before</u> SWAU	<a href="#">AP-341-519</a> change documents <sup>2</sup>	R	R	R	G	-	-	-	-
Minor or major changes made once software is baselined	Revised baseline <sup>3</sup>	R	R	R	-	R	R	R	-
Minor or major changes made <u>after</u> SWAU	SSC Software Change Package (SCP) Form <a href="#">AP-341-507-FM01</a> <sup>4</sup>	R	R	G	-	-	-	-	-
	Non-SSC SWNCP Form, SOFT-GEN-FM02	-	-	-	-	R	R	G	-
<p><sup>1</sup> See Form 2033 for a crosswalk between MLs and software type, category and software risk levels. R = Required. G = Required but graded. "-" = Not required.</p> <p><sup>2</sup> Includes Field Change Notices (FCNs), Field Change Requests (FCRs), Design Revision Notices (DRNs), Design Change Forms (DCFs) as described in <a href="#">AP-341-519</a> unless explicitly superseded by another administrative procedure.</p> <p><sup>3</sup> Use SOFT-GEN-FM01, <i>Software Baseline Form (SWBL)</i>. Include documents and computer program files.</p> <p><sup>4</sup> See ESM Chapter 15, <i>Commissioning</i> and <a href="#">AP-341-801</a>, <i>Post Modification/Post Maintenance Testing</i> for related testing requirements of software changes.</p>									

This search tool may not include the latest files; use individual files for work.

- d. Perform appropriate verification activities and validation (V&V) of changes (see SOFT-V&V). Perform appropriate acceptance testing for the change.
- d. Incorporate the changes in documentation, and maintain traceability of the change to the software design requirements.
  - 1) Include the (1) initiation, evaluation, and disposition of a change request, (2) control and approval of changes prior to implementation, and (3) requirements for retesting (e.g., regression testing) and acceptance of the test results.
- e. Perform the following:
  - 1) maintain a current status of configuration items and control configuration changes until they are incorporated into the approved product baseline;



## 1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

- 2) maintain the status of changes that are proposed and approved, but not implemented; and
  - 3) provide notification of this information to affected organizations.
  - f. For changes to documentation only, use the governing document control process.
  - g. Implement the processes described and/or referenced in this chapter to ensure that interfaces are controlled/managed such that unintended consequences do not occur.
  - h. As applicable, ensure configurations changes are established (Ref. [AP-341-613](#), *Instrumentation Set Point Control*).
  - i. Update the software baseline.
- B. When. Control changes per Table 21.1-4, in a timely manner, and when required to ensure proper performance.
- C. Who.
1. See [AP-341-507](#), [AP-341-519](#) unless explicitly superseded by another procedure and SOFT-GEN-FM02, Non-SSC Software Change Package Form (SWNCP).
  2. The organization responsible for the original software design and others as deemed necessary by the SRLM must evaluate, and approve software changes unless an alternate organization has been given the authority by the SRLM to approve the changes.
  3. As determined by the SRLM, only those that are knowledgeable in the computer program code may make changes to the computer program code.

**5.0 PERFORM PROBLEM REPORTING & CORRECTIVE ACTION**

- A. How.
1. For ML-1 through ML-4 software, perform software problem reporting and corrective action throughout the software life cycle.
  2. For ML-4 software, less formal methods (e.g., bug lists, use of comment fields within the computer program, software-tailored tools such as [TeamForge](#) and [trac](#)) or formal methods (see below) may be used throughout the software lifecycle.
  3. For ML-1 through ML-3 software, less formal methods may be used through development and testing; however, formal methods must be used after software approval for use (SWAU).<sup>22</sup> Individually or in combination, [P330-6](#), *Nonconformance Reporting*, [P322-4](#), *Laboratory Performance Feedback and Improvement Process*, or equivalent as determined by the SRLM, must address the following:
    - a. Describe the evaluation process for determining whether a reported problem is an error (see definitions) or other type of problem (e.g., user mistake).
    - b. Define the responsibilities for disposition of problem reports, including notification to the originator of the results of the evaluation.

<sup>22</sup> P330-6 r10 has little on software but r11 is expected to include more direction.

- c. When a problem is determined to be an error (see definitions), provide, as appropriate, for
    - 1) how the error relates to appropriate software engineering elements;
    - 2) how the error impacts past and present use of the computer program;
    - 3) how the corrective action impacts previous development activities; and
    - 4) how the users are notified of the identified error, its impact, and how to avoid the error, pending implementation of corrective actions.
  - d. Problem reporting and corrective action must include methods for documenting, evaluating, and correcting software problems.
  - e. Conditions adverse to quality shall be identified promptly and corrected as soon as practicable.
  - f. In the case of a significant condition adverse to quality, the cause of the condition shall be determined and corrective action taken to preclude recurrence.
  - g. The identification, cause, and corrective action for significant conditions adverse to quality shall be documented and reported to appropriate levels of management.
  - h. The verification of completed corrective actions.
- B. When. Perform throughout the software life cycle.
- C. Who. The SO performs software problem reporting and corrective action on behalf of the SRLM. The SRLM is responsible to ensure it is performed as required by the associated procedure (e.g., [P330-6](#) on NCRs).

## 6.0 PERFORM SOFTWARE (PROJECT) RISK MANAGEMENT

- A. How.
1. Software risk management as required by O414.1D focuses on the risks to successfully complete the software project; it does not focus on the risks of potential failure of the software.<sup>23</sup> Software risk management applies to all phases of the software project life cycle. Overall software risk management is achieved through implementation of the NQA-1 based, systematic processes of this chapter and the supporting quality assurance program. That is, successful implementation of this chapter should result in successful completion of a software project. The SRLM should determine if project-level software Risk Management is needed (e.g., large and important projects).
  2. For ML-1 through ML-3 software, also address software-specific risks (risks specific or unique to the software implementation). Manage software-specific risks by (a) assessing and (b) controlling the risks.
    - a. With a graded approach approved by the SRLM and Chapter POC, implement risk management based on industry accepted methods such as those

<sup>23</sup> Based on [DOE G 414.1-4](#), *Safety Software Guide for Use with 10 CFR 830 Subpart A, Quality Assurance Requirements*, and [DOE O 414.1C](#), *Quality Assurance*.

described in [DOE SQAS21.01.00-1999](#), *Software Risk Management a Practical Guide*; [DOE G413.3-7](#), *Risk Management Guide*; and/or [DOE G 414.1-4](#), *Safety Software Guide for Use with 10 CFR 830 Subpart A, Quality Assurance Requirements, and DOE O 414.1C, Quality Assurance*.

3. *Risk assessment includes risk identification, analysis, and prioritization to ensure that the necessary resources are available to mitigate them.*
  4. *Risk control includes risk resolution and tracking. Resolve risks using risk avoidance, mitigation and/or transference.*
  5. Document and maintain software-specific risks on a risk list (also referred to as risk register) in the SWDS.
- B. When. Perform throughout the software life cycle.
- C. Who. The SO performs software risk management on behalf of the SRLM. The SRLM is responsible to ensure it is performed, and reviews and approves the risk list as part of the SWDS.

## 7.0 APPENDICES, ATTACHMENTS AND FORMS

### Appendices:

- APPENDIX A: Chapter 21 Definitions and Acronyms
- APPENDIX B: SO and SRLM Decision Diagram for FAO/COE
- APPENDIX C: Chapter 21 Reference Listing

### Attachments:

- ATTACHMENT 1: Chapter 21 Summary Table

### Forms:

- SOFT-GEN-FM01: Software Baseline Form (SWBL)
- SOFT-GEN-FM02: Non-SSC Software Change Package Form (SWNCP)
- SOFT-GEN-FM03: Software Datasheet Form (SWDS)

*This search tool may not include the latest files; use individual files for work.*

APPENDIX A: CHAPTER 21 DEFINITIONS AND ACRONYMS

Only key definitions are listed. See [PD340](#), *Conduct of Engineering for Facility Work*, [SD330](#), *Los Alamos National Laboratory Quality Assurance Program*, and documents referenced therein for additional definitions.

Table 21.1-A1 Chapter 21 Definitions	
Item	Definition
acceptance testing, also known as software validation	The process of exercising or evaluating a system or system component by manual or automated means to ensure that it satisfies the specified requirements, and, to identify differences between expected and actual results in the operating environment. (Ref. <a href="#">NQA-1</a> ).
acquired software	Software that is generally supplied through basic procurement, two-party agreements, or other contractual arrangements. Acquired software includes off-the-shelf (OTS) software such as operating systems, database management systems, compilers, software development tools, and commercial calculational software and spreadsheet tools. Downloadable software that is available at no cost to the user (referred to as freeware) is also considered acquired software. (Based on <a href="#">DOE G 414.1-4</a> ).
Acquired Non-SSC software	Non-SSC software that is acquired software where the code cannot be changed other than through replacement. This may also be referred to as Read-Only Non-SSC software. Replacement includes replacement with a subsequent software version or upgrade. (Definition developed for this chapter).
baseline	A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for use and further development, and that can be changed only by using an approved change control process. (Ref. <a href="#">NQA-1</a> ). <b>Note:</b> See SOFT-GEN-FM01, <i>Software Baseline Form (SWBL)</i> . A software baseline includes the computer program (code and [configuration] data) and the computer program documentation. In layman's terms, one may think of this as the information and computer program files that are needed to run the software for a specific application.
Change - Major computer program change	A change or bug fix that: <ul style="list-style-type: none"> <li>▪ the Software Responsible Line Manager (SRLM) or computer program supplier designates as a major change,</li> <li>▪ adds or deletes an ML-1, ML-2 or ML-3 SSC performance function (including set point changes),</li> <li>▪ modifies ML-1 or ML-2 SSC performance function code, excluding clarifying notes,</li> <li>▪ adds, deletes or modifies design and/or analysis output values of ML-1, ML-2 or ML-3 calculations,</li> <li>▪ recodes to another language, or</li> <li>▪ modifies a significant number of lines of code.</li> </ul> <b>Note:</b> A Major OTS computer program change is often indicated with increment increase in version number (e.g., change from version 1 to 2). An evaluation of the software however, is required to determine whether the version release is a major change. <b>Examples:</b> A change from Delta V control system software from version 7.0 to version 8.0. A change that adds code to implement an interlock functional performance requirement that an ML-3 laser system cannot be activated until area doors are locked. A change that modifies code on ML-2 ventilation system backdraft damper so that damper closure does not slam shut and potentially damage the damper assembly. A change in the algorithm or databased used for calculating the water flow rate in an ML-3 fire protection piping system design. A change in coding language from C to C++. A version change where 40% of the lines of code were modified. (Definition developed for this chapter).

Table 21.1-A1 Chapter 21 Definitions	
Item	Definition
Change - Minor computer program change	<p>A change or bug fix that is not a major computer program change but:</p> <ul style="list-style-type: none"> <li>adds or deletes an ML-4 SSC performance function (including setpoint changes),</li> <li>modifies ML-3 SSC performance function code, excluding clarifying notes, or</li> <li>adds, deletes or modifies design and/or analysis output values of ML-4 calculations</li> </ul> <p><b>Note:</b> A minor OTS software change is often indicated with a fractional increase in version number (e.g., 1.1 or 1.01). An evaluation of the software however, is required to determine whether the version release is a minor change.</p> <p><b>Examples:</b> Add code to implement automatic pump shut-off performance requirement on ML-4 sump low-level alarm. Modify code to fix a coding error on an ML-3 heating/cooling system so that cooling, rather than heating activates at high temperatures. Change the algorithm for calculating the current that flows in an ML-4 electric power system under abnormal conditions. (Definition developed for this chapter).</p>
Change - Less than minor computer program change	<p>A change or bug fix that is not a major or minor computer program change but:</p> <ul style="list-style-type: none"> <li>adds, deleted and/or modifies ML-4 performance function code,</li> <li>adds, deletes and/or modifies code that does <u>not</u> modify ML-1, ML-2, ML-3 performance function, or</li> <li>imparts changes without adding, deleting or modifying design and/or analysis output values (all MLs).</li> </ul> <p><b>Examples:</b> Modify code to increase the ramp time on an ML-4 softstart pump. Install security patch/service pack updates. An OTS software patch that includes a code change to prevent a screen from “freezing” or loading slowly (all MLs). Add/modify code clarifying notes (all MLs). Modify code to produce multiple reporting formats (all MLs). (Definition developed for this chapter).</p>
commercially controlled software	<p>As determined using <a href="#">Form 2033, Safety/Non-Safety Software Determination, Categorization, and Software Risk Level (SRL)</a>, software that is not, or will not be safety software or risk significant software. Such software may be acquired (including off the shelf (OTS) software, freeware, or designed software).</p> <p><b>Examples:</b> Personal productivity software (e.g., Microsoft PowerPoint, Oracle Project Primavera, MS Outlook, etc.); typical business accounting systems, facility personnel comfort temperature control systems. (Based on <a href="#">P1040</a>).</p>
computer program	<p>A combination of computer instructions and data definitions that enables computer hardware to perform computational or control functions. (Ref. <a href="#">NQA-1</a>).</p>
computer program listings	<p>A printout or other human readable display of the source and, sometimes, object statements that make up a computer program. (Ref. <a href="#">ISO/IEC/IEEE 24765</a>).</p>
configuration management	<p>The process of identifying and defining the configuration items in a system (i.e., software and hardware), controlling the release and change of these items throughout the system’s life cycle, and recording and reporting the status of configuration items and change requests. (Ref. <a href="#">NQA-1</a>).</p>
Designed (or developed) software	<p>Software that is designed or developed rather than acquired.</p> <p><b>Note:</b> Includes the following from <a href="#">DOE G 414.1-4</a>. (a) custom developed software, (b) configurable software, and (c) utility calculation software. (Definition developed for this chapter).</p>
Designed Non-SSC Software	<p>Non-SSC software where the computer program can be changed other than through replacement. Replacement includes replacement with a subsequent software version or upgrade. (Definition developed for this chapter).</p>
Designed SSC Software	<p>SSC software where the computer program can be changed other than through replacement of the computer program and/or the associated SSC. (Definition developed for this chapter).</p>

## 1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

Table 21.1-A1 Chapter 21 Definitions	
Item	Definition
Engineering Services Software Coordinator	Individual assigned to maintain the software inventory for ES Division, and to perform other duties as assigned (e.g., per desktop instruction [future]).
error	An error is a condition deviating from an established baseline including deviations from the current approved computer program and its baseline requirements. (Ref. <a href="#">NQA-1</a> ). An error is something which requires a software change (major or minor). Examples of errors include (a) if a computer program used for design of a structural member provides incorrect design output, and (b) if a computer program turns on heating instead of cooling at high temperature settings.
freeware	Software that is available for use at no cost or for a nominal, usually voluntary fee. (Ref. <a href="#">Merriam-Webster Dictionary</a> ).
major document change	A document change that is not a minor document change. A major document change includes revisions, changes, or modifications to a document (e.g., procedure, work instruction, drawing, etc.) which impact the effective implementation of the requirement(s). (Based on <a href="#">P1020-2</a> ).
minor document change	A document change, as defined by the governing document control program, that includes but is not limited to inconsequential editorial corrections, grammatical and spelling changes, organizational name and acronym changes, and similar type changes. (Based on <a href="#">P1020-2</a> ).
model	Simplifications of the real world constructed to gain insights into select attributes of a particular physical, biological, economic, engineered, or social system. (Ref. EPA/100K-09/003, <i>Guidance on the Development, Evaluation, and Application of Environmental Models</i> .)
Non-safety software	As determined using <a href="#">Form 2039</a> , software that is not otherwise determined to be safety software. Non-safety software includes risk significant and commercially controlled software. (Ref. <a href="#">P1040</a> ).
Non-SSC software	Software used in design, analysis and/or for administrative control. This software does not physically monitor and/or control SSCs. <b>Examples:</b> Piping system design/analysis software (CAESAR II®), fire protection system design software (SprinkCAD), area lighting calculation software, spreadsheets used to perform structural load calculations, safety analysis software used to perform dispersion modeling, software used to track facility combustible loading, and software used to track Technical Safety Requirement (TSR) implementation. (Definition developed for this chapter)
operating environment	A collection of software, firmware, and hardware elements that provide for the execution of computer programs. (Ref. <a href="#">NQA-1</a> ). It is also the location and conditions (environment) where the software will be used or operated to meet its intended function. (Based on <a href="#">P330-8</a> ).
Read-Only SSC Software	SSC software where the computer program cannot be changed other than through replacement of the computer program and/or the associated SSC. (It is sometimes referred to as embedded software or firmware; however, for this Chapter, the term Read-Only SSC software is used.) <b>Note:</b> Read-only software includes software where limits and/or set points can be configured (e.g., via keypad entries) without modifying the computer program (code). (Definition developed for this chapter).

1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

Table 21.1-A1 Chapter 21 Definitions	
Item	Definition
Risk Significant software	Software that is, or will be, used for any of the purposes for which safety software is used (see <a href="#">Form 2033</a> ), only such purposes are in or for an accelerator, live-fire range, biological hazard facility, explosive hazard facility, or a moderate- or high- hazard chemical facility; or failure of the software would prevent LANL from performing Essential Functions as described in <a href="#">EPP-COOP-006</a> , <i>LANL COOP [Continuity of Operations] Plan</i> . Commercially controlled software is not risk significant software. (Ref. <a href="#">P1040</a> ).
Safety and Hazard Analysis and Design Software (SHADS)	Safety software that is used, or will be used, to classify, design, or analyze nuclear (including radiological) facilities. This software is not part of an SSC, but helps to ensure the proper accident or hazards analysis of nuclear (including radiological) facilities or an SSC that performs a safety function. This is Non-SSC software, safety software, and is categorized as SHADS. (Ref. <a href="#">P1040</a> ).
Safety Management and Administrative Controls Software (SMACS)	Safety software that performs, or will perform, a hazard control function in support of nuclear (including radiological) facility radiological Safety Management Programs (SMPs) or TSRs; and/or this is software that performs, or will perform, a control function in support of a nuclear (including radiological) facility necessary to provide adequate protection from nuclear (including radiological) facility radiological hazards. It supports eliminating, limiting, or mitigating nuclear hazards to workers, the public, or the environment as addressed in <a href="#">10 CFR 830</a> , <a href="#">10 CFR 935</a> , <i>Occupational Radiation Protection</i> , and the Department of Energy Acquisition Regulation (DEAR) Integrated Safety Management System (ISMS) clause <a href="#">48 CFR 970.5223-1</a> , <i>Integration of Environment, Safety, and Health into Work Planning and Execution</i> . This is safety software and is categorized as SMACS. (this is a clarification of definition in <a href="#">P1040</a> r9; r10 expected to include same).
Safety Software	Software that includes any of the following: SSS, SHADS, or SMACS. Both SSC software and Non-SSC software can be safety software. (Ref. <a href="#">P1040</a> ).
Safety System Software (SSS)	Safety software for a nuclear (including radiological) facility that performs, or will perform, a safety function as part of an SSC and is cited in either (a) a DOE-approved documented safety analysis, or (b) an approved hazard analysis per <a href="#">DOE P 450.4A</a> , <i>Integrated Safety Management Policy</i> and <a href="#">48 CFR 970-5223-1</a> , <i>Integration of Environment, Safety, and Health into Work Planning and Execution</i> . This is SSC software, safety software and is categorized as SSS.  <b>Note:</b> References implemented at LANL as described in PD110, <i>Safety Basis</i> . See DOE-approved documented safety analyses at <a href="#">LANL Safety Basis Document Lists (SBDLs)</a> . Analyses include Documented Safety Analyses (DSAs), Preliminary Documented Safety Analyses (PDSAs), Bases for Interim Operations (BIOs) etc. (Based on <a href="#">P1040</a> ).
simple and easily understood (Non-SSC) software used in the design of SSCs	Software that satisfies the following criteria: a. The software is used in the design of SSCs; b. The results of the computer program can be easily confirmed through hand calculations; c. A person technically qualified in the subject can review and understand the program and the supporting calculations; and, d. The software can be individually verified with each use (e.g., calculation). (Based on <a href="#">NQA-1</a> ).
software	Computer programs and associated documentation and data pertaining to [needed for] the operation of a computer system. (Ref. <a href="#">NQA-1</a> ).
software approve/approval /approved for use (SWAU)	An approval that constitutes that the software requirements have been satisfied (including installation and operating instructions), and the software is ready to be used in the intended operating environment. (Ref. <a href="#">P1040</a> ).

## 1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

Table 21.1-A1 Chapter 21 Definitions	
Item	Definition
Software Coordinator	Individual(s), designated by division management, providing coordinating and/or administrative functions in support of chapter compliance (e.g., inventory and associated reporting). <i>ES Division has an ES-Div Software Coordinator.</i>
software design requirement	A requirement that impacts or constrains the design of a software system or software system component. (Based on <a href="#">ISO/IEC/IEEE 24765</a> ).
software engineering elements	(a) software acquisition method(s) for controlling the acquisition process for software and software services; (b) software engineering method(s) used to manage the software life-cycle activities; (c) application of standards, conventions, and other work practices that support the software life cycle; (d) controls for support software used to develop, operate, and maintain computer programs. (Ref. <a href="#">NQA-1</a> ).
Software Owner (SO)	Selected responsibilities from P1040 Rev 9 follow; see others in Ch 21: <ul style="list-style-type: none"> <li>Provides the software information and <a href="#">Form 2033, Safety/Non-Safety Software Determination, Categorization, and Software Risk Level (SRL)</a> and obtains review and concurrence of the form in accordance with this document.</li> <li>Reviews and approves the software project planning documentation.</li> <li>Owens the software and supports the SRLM in complying with the requirements of this document.</li> <li>Prepares the approval for use documentation that describes the intended use and any associated limitations, access controls, etc., for using the software.</li> </ul>
Software Point of Contact (SPOC)	That individual selected by division management to act as software owner for specific software or multiple Non-SCC software programs. Ideally it's the main or super-user of each program, but can be another user or even a single individual for a group or division.

This search tool may not include the latest files. Use individual files to work.



Table 21.1-A1 Chapter 21 Definitions	
Item	Definition
Software Responsible Line Manager (SLRM)	<p>Responsibilities (performs or causes to be performed, from P1040 Rev 9 for Software Owner RLM follow; see others in Ch 21:</p> <ul style="list-style-type: none"> <li>Manages and maintains software in accordance with this document to ensure it operates as intended.</li> <li>Determines reasonable probability, software type, software category, and as applicable SRL.</li> <li>Reviews Form 2033, Safety/Non-Safety Software Determination, Categorization, and Software Risk Level (SRL) for completeness, accuracy and adequacy, and signs Form 2033 indicating concurrence with the software determination, categorization, and as applicable, SRL.</li> <li>Retains the completed Form 2033 as a record in accordance with the SRLM's records management system.</li> <li>Identifies, documents, approves, controls, and maintains safety and risk significant software owned by the SRLM that is new software or in-use software at LANL nuclear (including radiological) facilities.</li> <li>Provides SRLM safety software inventory information and any changes thereto to QPA-IQ.</li> <li>Transmits the SRLM inventory information to QPA-IQ in accordance with applicable document control protocols, including as required, derivative classification review and marking for sensitive information; retains information as a record.</li> <li>Applies the appropriate amount of SQA rigor in software planning and implementation.</li> <li>Reviews and approves the software project planning documentation.</li> <li>As applicable, acquires software and/or software services.</li> <li>Except for SQA associated with using the software, is responsible for the SQA of the software.</li> <li>V&amp;Vs the software.</li> <li>Identifies and ensures reviews are performed by competent individuals or groups other than those who developed and documented in the original software design (but who may be from the same organization).</li> <li>Ensures software owning organization personnel managing or working to this document are adequately trained and as required, qualified.</li> <li>Approves software for use.</li> <li>Completes in-use tests in the operating environment.</li> <li>Retires software, including removal of safety software from software inventory.</li> </ul>
software tool	<p>A computer program used in the development, testing, analysis or maintenance of a program or its documentation. Examples include vendor-supplier configuration tools, conversion tables, comparators, cross-reference generators, compilers, CASE (Computer-Aided Design Software Engineering) tools, configuration and code management software, decompilers, disassemblers, editors, flowcharters, monitor test case generators, and timing analyzers. (Based on <a href="#">NQA-1</a>).</p>
Software User (SU)	<p>Responsibilities from P1040 Rev 9; see others in Ch 21.</p> <ul style="list-style-type: none"> <li>Reviews and approves the software project planning documentation.</li> <li>Applies the appropriate amount of SQA rigor in software planning and implementation.</li> <li>Reports software errors and problems.</li> <li>Uses software within software limitations and in accordance with this document.</li> </ul> <p>[Note, for SSC software, choose as the SU the individual that best adds value when considering the above role. <i>It may depend on and change whether in design or an operating facility. It may be the responsible engineer (or technician or maintenance in some cases) rather than facility ops personnel.</i>]</p>

1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

Table 21.1-A1 Chapter 21 Definitions	
Item	Definition
Software User Responsible Line Manager (SU RLM)	<p>Selected responsibilities from P1040 Rev 9; see others in Ch 21.</p> <ul style="list-style-type: none"> <li>• Reviews and approves the software project planning documentation.</li> <li>• Approves software for use.</li> <li>• Supports completion of in-use tests in the operating environment.</li> <li>• Ensures software users and software user organization personnel managing or working to this document are adequately trained, and as required, qualified.</li> </ul>
SSC software	<p>Software that controls and/or monitors system, structures and components (SSCs) and is running and interacting with its environment in real time. SSC software may be safety or non-safety software.</p> <p><u>Examples:</u> Building Automation Control System (BAS) software, process gas monitoring and control system software, fire alarm control panel (FACP) software, continuous air monitor (CAM) software, seismic switch software, and uninterrupted power supply (UPS) software. (Based on TR. No. 397, <i>Quality Assurance for Software Important to Safety</i>, IAEA, 2000).</p>
support software	<p>Software or a program that aides in the development, maintenance, or use of other software or provides general application-independent capability (Ref. <a href="#">ISO/IEC/IEEE 24765</a>). Support software includes software tools and system software (Ref. <a href="#">NQA-1</a>).</p> <p><b>Note:</b> SSC and Non-SSC software may have support software.</p>
system software	<p>An element of support software, the computer programs used to provide basic or general functionality and facilitate the operation and maintenance of the application computer program. Examples include lower level software layers, assemblers, interpreters, diagnostics, and utilities. (Based on <a href="#">NQA-1</a>).</p>
test case	<p>A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. (Ref. <a href="#">NQA-1</a>).</p>
test plan (Procedure)	<p>A document that describes the approach to be followed for testing a system or component. Typical contents identify the items to be tested, tasks to be performed, and responsibilities for the testing activities. (Ref. <a href="#">NQA-1</a>).</p>
toolbox code	<p>Software that is listed on the <a href="#">DOE Safety Software Quality Assurance Central Registry</a> (Registry). (Ref. <a href="#">P1040</a>).</p>
validation (software)	<p>The process of exercising or evaluating a system or system component by manual or automated means to ensure that it satisfies the specified requirements and to identify differences between expected and actual results in an operating environment (Ref. <a href="#">NQA-1</a>); and providing evidence that the software, and its associated products, satisfies system requirements allocated to software at the end of each life cycle activity, solves the right problem (e.g., correctly models physical laws, implements business rules, uses the proper system assumptions), and satisfies the intended use and user needs (Ref. <a href="#">DOE O 414.1D</a>).</p>
verification	<p>The act of reviewing, inspecting, testing, checking, auditing, or otherwise determining and documenting whether items, processes, services, or documents conform to specified requirements (Ref. <a href="#">NQA-1</a>); and providing objective evidence that the software and its associated products conform to requirements (e.g., for correctness, completeness, consistency, and accuracy) for all life cycle activities during each life cycle process (e.g., acquisition, supply, development, operation, and maintenance); satisfy standards, practices, and conventions during life cycle processes; successfully complete each life cycle activity; and satisfy all the criteria for initiating succeeding life cycle activities (e.g., building the software correctly) (Ref. <a href="#">O 414.1D</a>).</p>

## 1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

Table 21.1-A2 Chapter 21 Acronyms	
Acronym <sup>1</sup>	Definition
A	Approve
AFU	Approval for Use
ASCE	American Society of Civil Engineers
AP	Administrative Procedure
ASM	Acquisition Services Management
ASME	American Society of Mechanical Engineers
BAS	Building Automation System
BIO	Basis for Interim Operations
CAM	Continuous Air Monitor
CGD	Commercial Grade Dedication (Dedicated)
CM	Configuration Management
COOP	Continuity of Operations Plan
Cx	Commissioning
D	Develop
DA	Design Authority
DAG	Design Agency
DCF	Design Change Form
DOE	(United States) Department of Energy
DRN	Design Revision Notice
DSA	Documented Safety Analysis
EF	Essential Function
ES	Engineering Services
ES-Div	Engineering Services Division
ESM	Engineering Standards Manual
ESSC	Engineering Services Software Coordinator
FAC-COE	Facility Conduct of Engineering
FACP	Fire Alarm Control Panel
FCR	Field Change Request
FDAR	Facility Design Authority Representative
FDD	Facility Design Description
G	Grade (or Guide)
IEEE	Institute of Electrical and Electronics Engineers
IESL	LANL Institutional Evaluated Supplier List
IWR	Interim Work Request
LANL	Los Alamos National Laboratory
LCxA	LANL Commissioning Authority
ML	Management Level
NA	Not Applicable
NQA-1	<a href="#">ASME NQA-1-2008/NQA-1A-2009</a> , Quality Assurance Requirements for Nuclear Facility Applications, Part I and Part II
OCIO	Office of Chief Information Officer
O&M	Operations and Maintenance

Table 21.1-A2 Chapter 21 Acronyms	
Acronym <sup>1</sup>	Definition
OTS	Off the Shelf
PDSA	Preliminary Documented Safety Analysis
PFD	Process Flow Diagram
P&ID	Process and Instrumentation Diagram
POC	Point of Contact
R	Required or review (see context)
RE	Responsible Engineer
RLM	Responsible Line Manager
SB	Safety Basis Division
SBP	Safety Basis Procedure
S/CI	Suspect/Counterfeit Item
SDD	System Design Description
SHADS	Safety and Hazard Analysis Software
SMACS	Safety Management and Administrative Controls Software
SME	Subject Matter Expert
SO	Software Owner
SOO	Sequence of Operations
SOW	Statement of Work
SPOC	Software Point of Contact
SRLM	Software Responsible Line Manager
SSC	Structure, System Component
SSS	Safety System Software
SRL	Software Risk Level
SU	Software User
SWAU	Software Approve (Approval) for Use
SWDD	Software Design Documentation (or Document)
SWBL	Software Baseline
SWDS	Software Data Sheet
SWHA	Software Hazards Analysis
SWID	Software Identification Number
SWNCP	Non-SSC Software Change Package
SWRS	Software Requirements Specification
SWYRS	Software System Requirements Specification
SWTM	Software Traceability Matrix
SWTP	Software Test Plan
SWTR	Software Test Report
TA	Technical Area
UPS	Uninterrupted Power Supply
V&V	Verify and Validate (or verification and validation)

<sup>1</sup> Only key acronyms are listed. See [PD340](#), *Conduct of Engineering for Facility Work* and documents referenced therein for additional meanings.

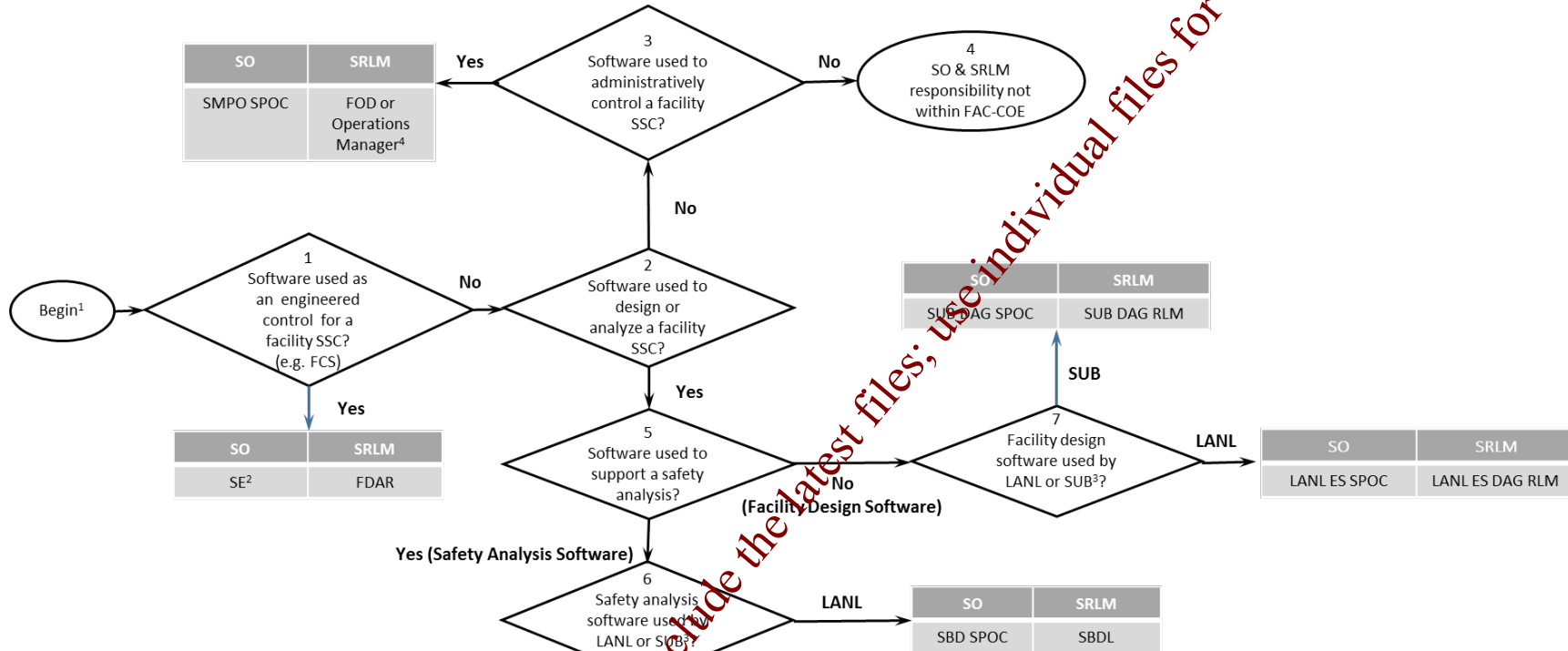
**APPENDIX B: SO AND SRLM DECISION DIAGRAM FOR FAC-COE**

Use Figure 21.1-B1 to identify the SO and SRLM. For clarification, contact the Facility Design Authority Representative (FDAR).

- For software used within a single facility/application and/or under a single FDAR, the associated FDAR has the ultimate authority to determine the appropriate SO and SRLM.
- For software used among multiple facilities/applications where there may be multiple FDARs, the LANL Facility Design Authority (DA) has the ultimate authority to make the determination.

*This search tool may not include the latest files; use individual files for work.*

Figure 21.1-B1 SO and SRLM Determination for FAC-COE (Guidance)



**NOTES:**

- <sup>1</sup> Answer the questions to determine the SO and SRLM.
- <sup>2</sup> Once operating, normally the plant system engineer, process engineer, or cognizant system engineer. During larger projects, functions are often performed by Design Agency.
- <sup>3</sup> As flowed down in supplier/subcontractor agreement.
- <sup>4</sup> For software used across multiple FODs, the SMP Program Owner should be the SRLM.
- <sup>5</sup> For clarification, contact the FDAR who has the ultimate authority to determine the appropriate SO and SRLM.

This search tool may not include the latest files; use individual files for work.

**KEY:**

DAG	Design Agency	SBDL	Safety Basis Division Leader
ES	Engineering Services Division	SE	System Engineer
FAC-COE	Facility Conduct of Engineering	SMPO	Safety Management Program Owner
FCS	Facility Control System	SO	Software Owner
FDAR	Facility Design Authority Representative	SPOC	Software Point of Contact
FOD	Facility Operating Director	SRLM	Software Responsible Line Manager
MAR	Material at Risk	SSC	System, Structure, Component
RLM	Responsible Line Manager	SUB	Subcontractor
SBA	Safety Basis Analyst		

## APPENDIX C: CHAPTER 21 REFERENCE LISTING

Most recent document revision (including revised document numbering and/or titles) applies unless otherwise stated. *Most national standards are available to LANL personnel via links; others must purchase.*

[10 CFR 830](#), Nuclear Safety Management

[10 CFR 835](#), Occupational Radiation Protection

[48 CFR 970-5223-1](#), Integration of Environment, Safety, and Health into Work Planning and Execution

[ANSI/ANS-10.7](#), Non-Real-Time, High-Integrity Software for the Nuclear Industry – Development Requirements

[ASME NQA-1-2008/NQA-1A-2009](#), Quality Assurance Requirements for Nuclear Facility Applications, Part I and Part II

[ASME NQA-1-2015 Part III, Subpart 3.2-2.14](#), Quality Assurance Requirements for Commercial Grade Items and Services, Commercial Grade Computer Programs, and Software Services

[Certified Software Quality Engineer Handbook](#), L. Westfall, American Society of Quality Pres.

## DOE

[DOE O 414.1D](#), Quality Assurance

[DOE Safety Software Quality Assurance Central Registry](#)

[DOE G 414.1-4](#), Safety Software Guide for Use with 10 CFR 830 Subpart A, Quality Assurance Requirements, and DOE O 414.1C, Quality Assurance (Revision A pending)

[DOE P 450.4A](#), Integrated Safety Management Policy

[DOE SQAS21.01.00-1999](#), Software Risk Management a Practical Guide

[DOE-STD-1195](#), Design of Safety Significant Safety Instrumented Systems Used at DOE Nonreactor Nuclear Facilities

[EPA/100/K-09/003](#), Guidance on the Development, Evaluation, and Application of Environmental Models, [www.epa.gov/crem](http://www.epa.gov/crem)

IAEA TR. No. 397, Quality Assurance for Software Important to Safety, IAEA, 2000

## IEEE

[ANSI/IEEE Std 7-4.3.2-2010](#), IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations

[ANSI/IEEE Std 828](#), IEEE Standard for Configuration Management in Systems and Software Engineering

[ANSI/IEEE Std 26514](#), Systems and Software Engineering—Requirements for Designers and Developers of User Documentation

[ANSI/IEEE Std 29148](#), Systems and Software Engineering – Life Cycle Processes – Requirements Engineering

[IEEE Std 344](#), Standard for Seismic Qualification of Equipment for Nuclear Power Generating Stations

[IEEE Std 829](#), IEEE Standard for Software and System Test Documentation

[IEEE Std 830](#), IEEE Recommended Practice for Software Specifications

[IEEE Std 1012](#), IEEE Standard for System and Software Verification

[IEEE Std 1016](#), IEEE Standard for Information Technology – System Design – Software Design Descriptions

[IEEE Std 1016.1](#), IEEE Guide to Software Design Descriptions

[IEEE Std 1028](#), IEEE Standard for Software Reviews and Audits

[IEEE Std 1666](#), IEEE Standard for Standard System C Language Reference Manual

[IEEE Std 12207](#), Systems and Software Engineering – Software Life Cycle Processes

[ISO/IEC/IEEE 24765](#), Systems and Software Engineering – Vocabulary

[ISO/IEC/IEEE 29148](#), Systems and Software Engineering-Life Cycle Processes-Requirements Engineering

[ANSI/ISA S84.00.01](#), Functional Safety: Safety Instrumented Systems for the Process Industry Sector

LANL (internal-only unless noted). Contact the chapter POC for assistance in obtaining LANL documents.

[AP-341-402](#), Engineering Document Management in Operating Facilities

[AP-341-405](#), Identification and Control of Technical Baseline, Variances, Alternate Methods, and Clarifications in Operating Facilities

## 1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

[AP-341-502](#), Management Level Determination and Identification of Quality Assurance and Maintenance Requirements

[AP-341-504](#), Temporary Modification Control

[AP-341-507](#), SSC Control Software Change Package

[AP-341-510](#), Field Walk-down, Data Gathering, and Inspections

[AP-341-519](#), Design Revision Control

[AP-341-605](#), Calculations

[AP-341-601](#), Functions and Requirements Document

[AP-341-602](#), Requirements and Criteria Document (RCD)

[AP-341-611](#), System Design Descriptions

[AP-341-613](#), Instrumentation Set Point Control

[AP-341-616](#), Technical Baseline Change During Design

[AP-341-620](#), Review of LANL Produced Design Documents

[AP-341-621](#), Design Authority Technical Review

[AP-341-622](#), LANL Review of Designs Produced by External Agencies

[AP-341-703](#), Commercial Grade Dedication

[AP-341-801](#), Post Modification/Post Maintenance Testing

[AP-341-802](#), System Health Reporting

[AP-341-901](#), Performing Vital Safety System Assessments

[AP-350-406](#), Startup and Commissioning

[ASM 3400.00.041](#), Goods or Services Requiring Special Review/Approval

[ASM Website](#)

[Conduct of Operations Resources](#)

[EPP-COOP-006](#), LANL COOP [Continuity of Operations] Plan

[ES Division Office \(ES-Div\) Website](#)

[Information System Security Officer \(ISSO\)](#)

[LANL Institutional Supplier List \(IESL\)](#)

[LANL Nuclear Facility List](#)

[LANL Safety Basis Document List](#)

[LA-14167-MS](#), Los Alamos National Laboratory, Concepts of Model Verification and Validation, Thacker et al., 2004

[P204-1](#), Controlled Unclassified Information

[P218](#), Cyber Security Access Controls

[P322-4](#), Laboratory Performance Feedback and Improvement Process

[P330-2](#), Control and Calibration of Measuring and Test Equipment (M&TE)

[P330-3](#), Quality Audits

[P330-5](#), Special Processes

[P330-6](#), Nonconformance Reporting

[P330-8](#), Inspection and Test

[P330-9](#), Suspect/Counterfeit Items (SCI)

[P341](#), Facility Engineering Processes Manual

[P342](#), Engineering Standards

[P343](#), Facility Engineering Training and Qualification Manual

[P805](#), Export Control

[P840-1](#), Quality Assurance for Procurements

[P1020-2](#), Laboratory Document Control

[P1040](#), Software Quality Management

[P1011](#), Computer Hardware, Software, and Services: Exceptions to Acquisition, Maintenance, and Support

[PD110](#), Safety Basis

[PD115](#), LANL Readiness Program

[PD210](#), Cyber Security Program

[PD328](#), LANL Assessment Program

[PD340](#), Conduct of Engineering for Facility Work

[PD370](#), Conduct of Engineering for Research and Development (R&D)

[QPA-DO-FSD-001](#), LANL Safety Software Inventory Procedure

[SBP-15-351](#), Design Basis or Safety Basis Change Review

[SBP111-1](#), Facility Hazard Categorization and Documentation

[SBP112-3](#), Unreviewed Safety Question (USQ) Process

[SBP113-3](#), Unreviewed Safety Issue (USI) Process

[SD210](#), Information Risk Management Framework

[SD330](#), Los Alamos National Laboratory Quality Assurance Program

[STD-342-100](#), Engineering Standards Manual (available externally)

[Merriam-Webster Dictionary](#)

[TeamForge \(LANL-internal\)](#)

[Trac](#)

APPENDIX A: CHAPTER 21 DEFINITIONS AND ACRONYMS

Only key definitions are listed. See [PD340](#), *Conduct of Engineering for Facility Work*, [SD330](#), *Los Alamos National Laboratory Quality Assurance Program*, and documents referenced therein for additional definitions.

Table 21.1-A1 Chapter 21 Definitions	
Item	Definition
acceptance testing, also known as software validation	The process of exercising or evaluating a system or system component by manual or automated means to ensure that it satisfies the specified requirements, and, to identify differences between expected and actual results in the operating environment. (Ref. <a href="#">NQA-1</a> ).
acquired software	Software that is generally supplied through basic procurement, two-party agreements, or other contractual arrangements. Acquired software includes off-the-shelf (OTS) software such as operating systems, database management systems, compilers, software development tools, and commercial calculational software and spreadsheet tools. Downloadable software that is available at no cost to the user (referred to as freeware) is also considered acquired software. (Based on <a href="#">DOE G 414.1-4</a> ).
Acquired Non-SSC software	Non-SSC software that is acquired software where the code cannot be changed other than through replacement. This may also be referred to as Read-Only Non-SSC software. Replacement includes replacement with a subsequent software version or upgrade. (Definition developed for this chapter).
baseline	A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for use and further development, and that can be changed only by using an approved change control process. (Ref. <a href="#">NQA-1</a> ). <b>Note:</b> See SOFT-GEN-FM01, <i>Software Baseline Form (SWBL)</i> . A software baseline includes the computer program (code and [configuration] data) and the computer program documentation. In layman's terms, one may think of this as the information and computer program files that are needed to run the software for a specific application.
Change - Major computer program change	A change or bug fix that: <ul style="list-style-type: none"> <li>▪ the Software Responsible Line Manager (SRLM) or computer program supplier designates as a major change,</li> <li>▪ adds or deletes an ML-1, ML-2 or ML-3 SSC performance function (including set point changes),</li> <li>▪ modifies ML-1 or ML-2 SSC performance function code, excluding clarifying notes,</li> <li>▪ adds, deletes or modifies design and/or analysis output values of ML-1, ML-2 or ML-3 calculations,</li> <li>▪ recodes to another language, or</li> <li>▪ modifies a significant number of lines of code.</li> </ul> <b>Note:</b> A Major OTS computer program change is often indicated with increment increase in version number (e.g., change from version 1 to 2). An evaluation of the software however, is required to determine whether the version release is a major change. <b>Examples:</b> A change from Delta V control system software from version 7.0 to version 8.0. A change that adds code to implement an interlock functional performance requirement that an ML-3 laser system cannot be activated until area doors are locked. A change that modifies code on ML-2 ventilation system backdraft damper so that damper closure does not slam shut and potentially damage the damper assembly. A change in the algorithm or databased used for calculating the water flow rate in an ML-3 fire protection piping system design. A change in coding language from C to C++. A version change where 40% of the lines of code were modified. (Definition developed for this chapter).



Table 21.1-A1 Chapter 21 Definitions	
Item	Definition
Change - Minor computer program change	<p>A change or bug fix that is not a major computer program change but:</p> <ul style="list-style-type: none"> <li>adds or deletes an ML-4 SSC performance function (including setpoint changes),</li> <li>modifies ML-3 SSC performance function code, excluding clarifying notes, or</li> <li>adds, deletes or modifies design and/or analysis output values of ML-4 calculations</li> </ul> <p><b>Note:</b> A minor OTS software change is often indicated with a fractional increase in version number (e.g., 1.1 or 1.01). An evaluation of the software however, is required to determine whether the version release is a minor change.</p> <p><b>Examples:</b> Add code to implement automatic pump shut-off performance requirement on ML-4 sump low-level alarm. Modify code to fix a coding error on an ML-3 heating/cooling system so that cooling, rather than heating activates at high temperatures. Change the algorithm for calculating the current that flows in an ML-4 electric power system under abnormal conditions. (Definition developed for this chapter).</p>
Change - Less than minor computer program change	<p>A change or bug fix that is not a major or minor computer program change but:</p> <ul style="list-style-type: none"> <li>adds, deleted and/or modifies ML-4 performance function code,</li> <li>adds, deletes and/or modifies code that does <u>not</u> modify ML-1, ML-2, ML-3 performance function, or</li> <li>imparts changes without adding, deleting or modifying design and/or analysis output values (all MLs).</li> </ul> <p><b>Examples:</b> Modify code to increase the ramp time on an ML-4 softstart pump. Install security patch/service pack updates. An OTS software patch that includes a code change to prevent a screen from “freezing” or loading slowly (all MLs). Add/modify code clarifying notes (all MLs). Modify code to produce multiple reporting formats (all MLs). (Definition developed for this chapter).</p>
commercially controlled software	<p>As determined using <a href="#">Form 2033, Safety/Non-Safety Software Determination, Categorization, and Software Risk Level (SRL)</a>, software that is not, or will not be safety software or risk significant software. Such software may be acquired (including off the shelf (OTS) software, freeware, or designed software).</p> <p><b>Examples:</b> Personal productivity software (e.g., Microsoft PowerPoint, Oracle Project Primavera, MS Outlook, etc.); typical business accounting systems, facility personnel comfort temperature control systems. (Based on <a href="#">P1040</a>).</p>
computer program	<p>A combination of computer instructions and data definitions that enables computer hardware to perform computational or control functions. (Ref. <a href="#">NQA-1</a>).</p>
computer program listings	<p>A printout or other human readable display of the source and, sometimes, object statements that make up a computer program. (Ref. <a href="#">ISO/IEC/IEEE 24765</a>).</p>
configuration management	<p>The process of identifying and defining the configuration items in a system (i.e., software and hardware), controlling the release and change of these items throughout the system’s life cycle, and recording and reporting the status of configuration items and change requests. (Ref. <a href="#">NQA-1</a>).</p>
Designed (or developed) software	<p>Software that is designed or developed rather than acquired.</p> <p><b>Note:</b> Includes the following from <a href="#">DOE G 414.1-4</a>. (a) custom developed software, (b) configurable software, and (c) utility calculation software. (Definition developed for this chapter).</p>
Designed Non-SSC Software	<p>Non-SSC software where the computer program can be changed other than through replacement. Replacement includes replacement with a subsequent software version or upgrade. (Definition developed for this chapter).</p>
Designed SSC Software	<p>SSC software where the computer program can be changed other than through replacement of the computer program and/or the associated SSC. (Definition developed for this chapter).</p>

1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

Table 21.1-A1 Chapter 21 Definitions	
Item	Definition
Engineering Services Software Coordinator	Individual assigned to maintain the software inventory for ES Division, and to perform other duties as assigned (e.g., per desktop instruction [future]).
error	An error is a condition deviating from an established baseline including deviations from the current approved computer program and its baseline requirements. (Ref. <a href="#">NQA-1</a> ). An error is something which requires a software change (major or minor). Examples of errors include (a) if a computer program used for design of a structural member provides incorrect design output, and (b) if a computer program turns on heating instead of cooling at high temperature settings.
freeware	Software that is available for use at no cost or for a nominal, usually voluntary fee. (Ref. <a href="#">Merriam-Webster Dictionary</a> ).
major document change	A document change that is not a minor document change. A major document change includes revisions, changes, or modifications to a document (e.g., procedure, work instruction, drawing, etc.) which impact the effective implementation of the requirement(s). (Based on <a href="#">P1020-2</a> ).
minor document change	A document change, as defined by the governing document control program, that includes but is not limited to inconsequential editorial corrections, grammatical and spelling changes, organizational name and acronym changes, and similar type changes. (Based on <a href="#">P1020-2</a> ).
model	Simplifications of the real world constructed to gain insights into select attributes of a particular physical, biological, economic, engineered, or social system. (Ref. EPA/100K-09/003, <i>Guidance on the Development, Evaluation, and Application of Environmental Models</i> .)
Non-safety software	As determined using <a href="#">Form 2039</a> , software that is not otherwise determined to be safety software. Non-safety software includes risk significant and commercially controlled software. (Ref. <a href="#">P1040</a> ).
Non-SSC software	Software used in design, analysis and/or for administrative control. This software does not physically monitor and/or control SSCs. <b>Examples:</b> Piping system design/analysis software (CAESAR II®), fire protection system design software (SprinkCAD), area lighting calculation software, spreadsheets used to perform structural load calculations, safety analysis software used to perform dispersion modeling, software used to track facility combustible loading, and software used to track Technical Safety Requirement (TSR) implementation. (Definition developed for this chapter)
operating environment	A collection of software, firmware, and hardware elements that provide for the execution of computer programs. (Ref. <a href="#">NQA-1</a> ). It is also the location and conditions (environment) where the software will be used or operated to meet its intended function. (Based on <a href="#">P330-8</a> ).
Read-Only SSC Software	SSC software where the computer program cannot be changed other than through replacement of the computer program and/or the associated SSC. (It is sometimes referred to as embedded software or firmware; however, for this Chapter, the term Read-Only SSC software is used.) <b>Note:</b> Read-only software includes software where limits and/or set points can be configured (e.g., via keypad entries) without modifying the computer program (code). (Definition developed for this chapter).

1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

Table 21.1-A1 Chapter 21 Definitions	
Item	Definition
Risk Significant software	Software that is, or will be, used for any of the purposes for which safety software is used (see <a href="#">Form 2033</a> ), only such purposes are in or for an accelerator, live-fire range, biological hazard facility, explosive hazard facility, or a moderate- or high- hazard chemical facility; or failure of the software would prevent LANL from performing Essential Functions as described in <a href="#">EPP-COOP-006</a> , <i>LANL COOP [Continuity of Operations] Plan</i> . Commercially controlled software is not risk significant software. (Ref. <a href="#">P1040</a> ).
Safety and Hazard Analysis and Design Software (SHADS)	Safety software that is used, or will be used, to classify, design, or analyze nuclear (including radiological) facilities. This software is not part of an SSC, but helps to ensure the proper accident or hazards analysis of nuclear (including radiological) facilities or an SSC that performs a safety function. This is Non-SSC software, safety software, and is categorized as SHADS. (Ref. <a href="#">P1040</a> ).
Safety Management and Administrative Controls Software (SMACS)	Safety software that performs, or will perform, a hazard control function in support of nuclear (including radiological) facility radiological Safety Management Programs (SMPs) or TSRs; and/or this is software that performs, or will perform, a control function in support of a nuclear (including radiological) facility necessary to provide adequate protection from nuclear (including radiological) facility radiological hazards. It supports eliminating, limiting, or mitigating nuclear hazards to workers, the public, or the environment as addressed in <a href="#">10 CFR 830</a> , <a href="#">10 CFR 935</a> , <i>Occupational Radiation Protection</i> , and the Department of Energy Acquisition Regulation (DEAR) Integrated Safety Management System (ISMS) clause <a href="#">48 CFR 970.5223-1</a> , <i>Integration of Environment, Safety, and Health into Work Planning and Execution</i> . This is safety software and is categorized as SMACS. (this is a clarification of definition in <a href="#">P1040</a> r9; r10 expected to include same).
Safety Software	Software that includes any of the following: SSS, SHADS, or SMACS. Both SSC software and Non-SSC software can be safety software. (Ref. <a href="#">P1040</a> ).
Safety System Software (SSS)	Safety software for a nuclear (including radiological) facility that performs, or will perform, a safety function as part of an SSC and is cited in either (a) a DOE-approved documented safety analysis, or (b) an approved hazard analysis per <a href="#">DOE P 450.4A</a> , <i>Integrated Safety Management Policy</i> and <a href="#">48 CFR 970-5223-1</a> , <i>Integration of Environment, Safety, and Health into Work Planning and Execution</i> . This is SSC software, safety software and is categorized as SSS.  <b>Note:</b> References implemented at LANL as described in PD110, <i>Safety Basis</i> . See DOE-approved documented safety analyses at <a href="#">LANL Safety Basis Document Lists (SBDLs)</a> . Analyses include Documented Safety Analyses (DSAs), Preliminary Documented Safety Analyses (PDSAs), Bases for Interim Operations (BIOs) etc. (Based on <a href="#">P1040</a> ).
simple and easily understood (Non-SSC) software used in the design of SSCs	Software that satisfies the following criteria: a. The software is used in the design of SSCs; b. The results of the computer program can be easily confirmed through hand calculations; c. A person technically qualified in the subject can review and understand the program and the supporting calculations; and, d. The software can be individually verified with each use (e.g., calculation). (Based on <a href="#">NQA-1</a> ).
software	Computer programs and associated documentation and data pertaining to [needed for] the operation of a computer system. (Ref. <a href="#">NQA-1</a> ).
software approve/approval /approved for use (SWAU)	An approval that constitutes that the software requirements have been satisfied (including installation and operating instructions), and the software is ready to be used in the intended operating environment. (Ref. <a href="#">P1040</a> ).

## 1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

Table 21.1-A1 Chapter 21 Definitions	
Item	Definition
Software Coordinator	Individual(s), designated by division management, providing coordinating and/or administrative functions in support of chapter compliance (e.g., inventory and associated reporting). <i>ES Division has an ES-Div Software Coordinator.</i>
software design requirement	A requirement that impacts or constrains the design of a software system or software system component. (Based on <a href="#">ISO/IEC/IEEE 24765</a> ).
software engineering elements	(a) software acquisition method(s) for controlling the acquisition process for software and software services; (b) software engineering method(s) used to manage the software life-cycle activities; (c) application of standards, conventions, and other work practices that support the software life cycle; (d) controls for support software used to develop, operate, and maintain computer programs. (Ref. <a href="#">NQA-1</a> ).
Software Owner (SO)	Selected responsibilities from P1040 Rev 9 follow; see others in Ch 21: <ul style="list-style-type: none"> <li>• Provides the software information and <a href="#">Form 2033, Safety/Non-Safety Software Determination, Categorization, and Software Risk Level (SRL)</a> and obtains review and concurrence of the form in accordance with this document.</li> <li>• Reviews and approves the software project planning documentation.</li> <li>• Owns the software and supports the SRLM in complying with the requirements of this document.</li> <li>• Prepares the approval for use documentation that describes the intended use and any associated limitations, access controls, etc., for using the software.</li> </ul>
Software Point of Contact (SPOC)	That individual selected by division management to act as software owner for specific software or multiple Non-SCC software programs. Ideally it's the main or super-user of each program, but can be another user or even a single individual for a group or division.

This search tool may not include the latest files. Use individual files to work.

Table 21.1-A1 Chapter 21 Definitions	
Item	Definition
Software Responsible Line Manager (SLRM)	<p>Responsibilities (performs or causes to be performed, from P1040 Rev 9 for Software Owner RLM follow; see others in Ch 21:</p> <ul style="list-style-type: none"> <li>• Manages and maintains software in accordance with this document to ensure it operates as intended.</li> <li>• Determines reasonable probability, software type, software category, and as applicable SRL.</li> <li>• Reviews Form 2033, Safety/Non-Safety Software Determination, Categorization, and Software Risk Level (SRL) for completeness, accuracy and adequacy, and signs Form 2033 indicating concurrence with the software determination, categorization, and as applicable, SRL.</li> <li>• Retains the completed Form 2033 as a record in accordance with the SRLM's records management system.</li> <li>• Identifies, documents, approves, controls, and maintains safety and risk significant software owned by the SRLM that is new software or in-use software at LANL nuclear (including radiological) facilities.</li> <li>• Provides SRLM safety software inventory information and any changes thereto to QPA-IQ.</li> <li>• Transmits the SRLM inventory information to QPA-IQ in accordance with applicable document control protocols, including as required, derivative classification review and marking for sensitive information; retains information as a record.</li> <li>• Applies the appropriate amount of SQA rigor in software planning and implementation.</li> <li>• Reviews and approves the software project planning documentation.</li> <li>• As applicable, acquires software and/or software services.</li> <li>• Except for SQA associated with using the software, is responsible for the SQA of the software.</li> <li>• V&amp;Vs the software.</li> <li>• Identifies and ensures reviews are performed by competent individuals or groups other than those who developed and documented in the original software design (but who may be from the same organization).</li> <li>• Ensures software owning organization personnel managing or working to this document are adequately trained and as required, qualified.</li> <li>• Approves software for use.</li> <li>• Completes in-use tests in the operating environment.</li> <li>• Retires software, including removal of safety software from software inventory.</li> </ul>
software tool	<p>A computer program used in the development, testing, analysis or maintenance of a program or its documentation. Examples include vendor-supplier configuration tools, conversion tables, comparators, cross-reference generators, compilers, CASE (Computer-Aided Design Software Engineering) tools, configuration and code management software, decompilers, disassemblers, editors, flowcharters, monitor test case generators, and timing analyzers. (Based on <a href="#">NQA-1</a>).</p>
Software User (SU)	<p>Responsibilities from P1040 Rev 9; see others in Ch 21.</p> <ul style="list-style-type: none"> <li>• Reviews and approves the software project planning documentation.</li> <li>• Applies the appropriate amount of SQA rigor in software planning and implementation.</li> <li>• Reports software errors and problems.</li> <li>• Uses software within software limitations and in accordance with this document.</li> </ul> <p>[Note, for SSC software, choose as the SU the individual that best adds value when considering the above role. <i>It may depend on and change whether in design or an operating facility. It may be the responsible engineer (or technician or maintenance in some cases) rather than facility ops personnel.</i>]</p>

Table 21.1-A1 Chapter 21 Definitions	
Item	Definition
Software User Responsible Line Manager (SU RLM)	<p>Selected responsibilities from P1040 Rev 9; see others in Ch 21.</p> <ul style="list-style-type: none"> <li>• Reviews and approves the software project planning documentation.</li> <li>• Approves software for use.</li> <li>• Supports completion of in-use tests in the operating environment.</li> <li>• Ensures software users and software user organization personnel managing or working to this document are adequately trained, and as required, qualified.</li> </ul>
SSC software	<p>Software that controls and/or monitors system, structures and components (SSCs) and is running and interacting with its environment in real time. SSC software may be safety or non-safety software.</p> <p><u>Examples:</u> Building Automation Control System (BAS) software, process gas monitoring and control system software, fire alarm control panel (FACP) software, continuous air monitor (CAM) software, seismic switch software, and uninterrupted power supply (UPS) software. (Based on TR. No. 397, <i>Quality Assurance for Software Important to Safety</i>, IAEA, 2000).</p>
support software	<p>Software or a program that aides in the development, maintenance, or use of other software or provides general application-independent capability (Ref. <a href="#">ISO/IEC/IEEE 24765</a>). Support software includes software tools and system software (Ref. <a href="#">NQA-1</a>).</p> <p><b>Note:</b> SSC and Non-SSC software may have support software.</p>
system software	<p>An element of support software, the computer programs used to provide basic or general functionality and facilitate the operation and maintenance of the application computer program. Examples include lower level software layers, assemblers, interpreters, diagnostics, and utilities. (Based on <a href="#">NQA-1</a>).</p>
test case	<p>A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. (Ref. <a href="#">NQA-1</a>).</p>
test plan (Procedure)	<p>A document that describes the approach to be followed for testing a system or component. Typical contents identify the items to be tested, tasks to be performed, and responsibilities for the testing activities. (Ref. <a href="#">NQA-1</a>).</p>
toolbox code	<p>Software that is listed on the <a href="#">DOE Safety Software Quality Assurance Central Registry</a> (Registry). (Ref. <a href="#">P1040</a>).</p>
validation (software)	<p>The process of exercising or evaluating a system or system component by manual or automated means to ensure that it satisfies the specified requirements and to identify differences between expected and actual results in an operating environment (Ref. <a href="#">NQA-1</a>); and providing evidence that the software, and its associated products, satisfies system requirements allocated to software at the end of each life cycle activity, solves the right problem (e.g., correctly models physical laws, implements business rules, uses the proper system assumptions), and satisfies the intended use and user needs (Ref. <a href="#">DOE O 414.1D</a>).</p>
verification	<p>The act of reviewing, inspecting, testing, checking, auditing, or otherwise determining and documenting whether items, processes, services, or documents conform to specified requirements (Ref. <a href="#">NQA-1</a>); and providing objective evidence that the software and its associated products conform to requirements (e.g., for correctness, completeness, consistency, and accuracy) for all life cycle activities during each life cycle process (e.g., acquisition, supply, development, operation, and maintenance); satisfy standards, practices, and conventions during life cycle processes; successfully complete each life cycle activity; and satisfy all the criteria for initiating succeeding life cycle activities (e.g., building the software correctly) (Ref. <a href="#">O 414.1D</a>).</p>

Table 21.1-A2 Chapter 21 Acronyms	
Acronym <sup>1</sup>	Definition
A	Approve
AFU	Approval for Use
ASCE	American Society of Civil Engineers
AP	Administrative Procedure
ASM	Acquisition Services Management
ASME	American Society of Mechanical Engineers
BAS	Building Automation System
BIO	Basis for Interim Operations
CAM	Continuous Air Monitor
CGD	Commercial Grade Dedication (Dedicated)
CM	Configuration Management
COOP	Continuity of Operations Plan
Cx	Commissioning
D	Develop
DA	Design Authority
DAG	Design Agency
DCF	Design Change Form
DOE	(United States) Department of Energy
DRN	Design Revision Notice
DSA	Documented Safety Analysis
EF	Essential Function
ES	Engineering Services
ES-Div	Engineering Services Division
ESM	Engineering Standards Manual
ESSC	Engineering Services Software Coordinator
FAC-COE	Facility Conduct of Engineering
FACP	Fire Alarm Control Panel
FCR	Field Change Request
FDAR	Facility Design Authority Representative
FDD	Facility Design Description
G	Grade (or Guide)
IEEE	Institute of Electrical and Electronics Engineers
IESL	LANL Institutional Evaluated Supplier List
IWR	Interim Work Request
LANL	Los Alamos National Laboratory
LCxA	LANL Commissioning Authority
ML	Management Level
NA	Not Applicable
NQA-1	<a href="#">ASME NQA-1-2008/NQA-1A-2009</a> , Quality Assurance Requirements for Nuclear Facility Applications, Part I and Part II
OCIO	Office of Chief Information Officer
O&M	Operations and Maintenance

Table 21.1-A2 Chapter 21 Acronyms	
Acronym <sup>1</sup>	Definition
OTS	Off the Shelf
PDSA	Preliminary Documented Safety Analysis
PFD	Process Flow Diagram
P&ID	Process and Instrumentation Diagram
POC	Point of Contact
R	Required or review (see context)
RE	Responsible Engineer
RLM	Responsible Line Manager
SB	Safety Basis Division
SBP	Safety Basis Procedure
S/CI	Suspect/Counterfeit Item
SDD	System Design Description
SHADS	Safety and Hazard Analysis Software
SMACS	Safety Management and Administrative Controls Software
SME	Subject Matter Expert
SO	Software Owner
SOO	Sequence of Operations
SOW	Statement of Work
SPOC	Software Point of Contact
SRLM	Software Responsible Line Manager
SSC	Structure, System Component
SSS	Safety System Software
SRL	Software Risk Level
SU	Software User
SWAU	Software Approve (Approval) for Use
SWDD	Software Design Documentation (or Document)
SWBL	Software Baseline
SWDS	Software Data Sheet
SWHA	Software Hazards Analysis
SWID	Software Identification Number
SWNCP	Non-SSC Software Change Package
SWRS	Software Requirements Specification
SWYRS	Software System Requirements Specification
SWTM	Software Traceability Matrix
SWTP	Software Test Plan
SWTR	Software Test Report
TA	Technical Area
UPS	Uninterrupted Power Supply
V&V	Verify and Validate (or verification and validation)

<sup>1</sup> Only key acronyms are listed. See [PD340](#), *Conduct of Engineering for Facility Work* and documents referenced therein for additional meanings.

**APPENDIX B: SO AND SRLM DECISION DIAGRAM FOR FAC-COE**

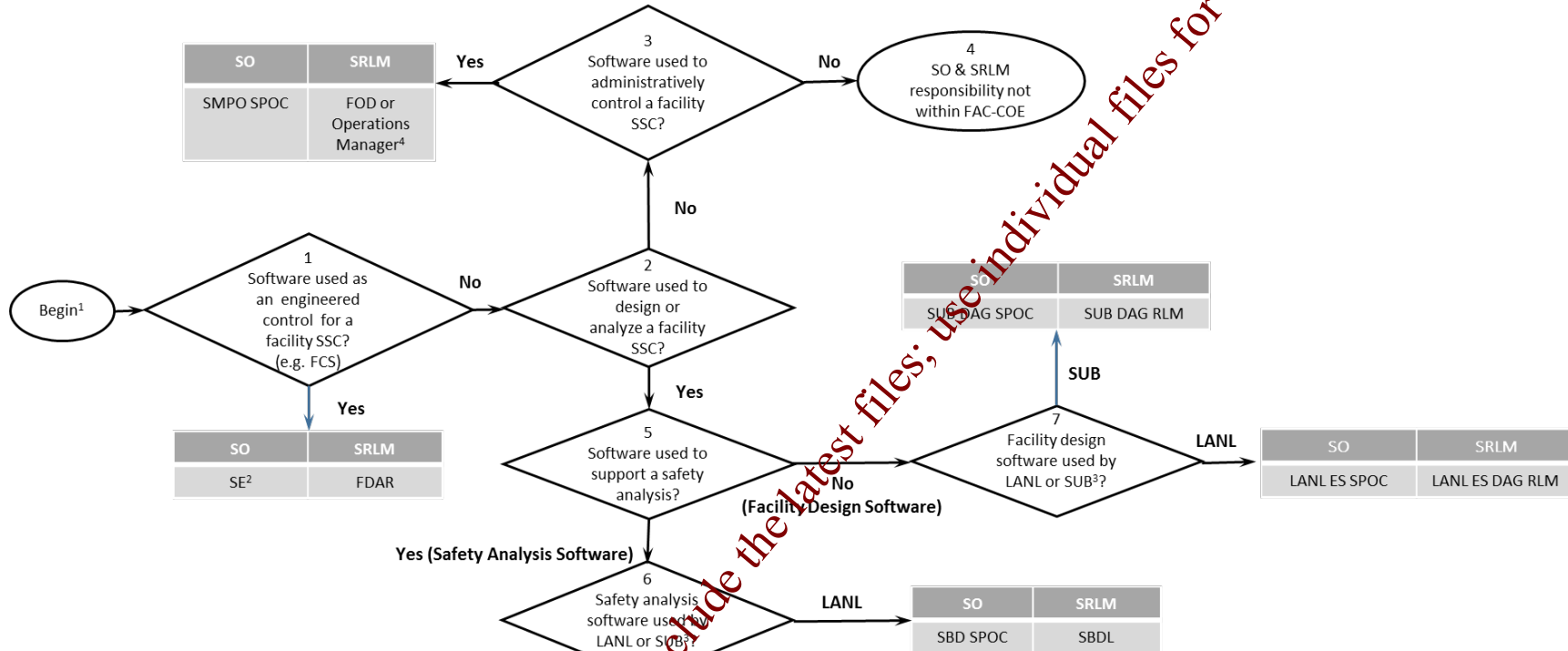
Use Figure 21.1-B1 to identify the SO and SRLM. For clarification, contact the Facility Design Authority Representative (FDAR).

- For software used within a single facility/application and/or under a single FDAR, the associated FDAR has the ultimate authority to determine the appropriate SO and SRLM.
- For software used among multiple facilities/applications where there may be multiple FDARs, the LANL Facility Design Authority (DA) has the ultimate authority to make the determination.

*This search tool may not include the latest files; use individual files for work.*



Figure 21.1-B1 SO and SRLM Determination for FAC-COE (Guidance)



**NOTES:**

- <sup>1</sup> Answer the questions to determine the SO and SRLM.
- <sup>2</sup> Once operating, normally the plant system engineer, process engineer, or cognizant system engineer. During larger projects, functions are often performed by Design Agency.
- <sup>3</sup> As flowed down in supplier/subcontractor agreement.
- <sup>4</sup> For software used across multiple FODs, the SMP Program Owner should be the SRLM.
- <sup>5</sup> For clarification, contact the FDAR who has the ultimate authority to determine the appropriate SO and SRLM.

**KEY:**

DAG	Design Agency	SBDL	Safety Basis Division Leader
ES	Engineering Services Division	SE	System Engineer
FAC-COE	Facility Conduct of Engineering	SMPO	Safety Management Program Owner
FCS	Facility Control System	SO	Software Owner
FDAR	Facility Design Authority Representative	SPOC	Software Point of Contact
FOD	Facility Operating Director	SRLM	Software Responsible Line Manager
MAR	Material at Risk	SSC	System, Structure, Component
RLM	Responsible Line Manager	SUB	Subcontractor
SBA	Safety Basis Analyst		

## APPENDIX C: CHAPTER 21 REFERENCE LISTING

Most recent document revision (including revised document numbering and/or titles) applies unless otherwise stated. *Most national standards are available to LANL personnel via links; others must purchase.*

[10 CFR 830](#), Nuclear Safety Management

[10 CFR 835](#), Occupational Radiation Protection

[48 CFR 970-5223-1](#), Integration of Environment, Safety, and Health into Work Planning and Execution

[ANSI/ANS-10.7](#), Non-Real-Time, High-Integrity Software for the Nuclear Industry – Development Requirements

[ASME NQA-1-2008/NQA-1A-2009](#), Quality Assurance Requirements for Nuclear Facility Applications, Part I and Part II

[ASME NQA-1-2015 Part III, Subpart 3.2-2.14](#), Quality Assurance Requirements for Commercial Grade Items and Services, Commercial Grade Computer Programs, and Software Services

[Certified Software Quality Engineer Handbook](#), L. Westfall, American Society of Quality Pres.

## DOE

[DOE O 414.1D](#), Quality Assurance

[DOE Safety Software Quality Assurance Central Registry](#)

[DOE G 414.1-4](#), Safety Software Guide for Use with 10 CFR 830 Subpart A, Quality Assurance Requirements, and DOE O 414.1C, Quality Assurance (Revision A pending)

[DOE P 450.4A](#), Integrated Safety Management Policy

[DOE SQAS21.01.00-1999](#), Software Risk Management a Practical Guide

[DOE-STD-1195](#), Design of Safety Significant Safety Instrumented Systems Used at DOE Nonreactor Nuclear Facilities

[EPA/100/K-09/003](#), Guidance on the Development, Evaluation, and Application of Environmental Models, [www.epa.gov/crem](http://www.epa.gov/crem)

IAEA TR. No. 397, Quality Assurance for Software Important to Safety, IAEA, 2000

## IEEE

[ANSI/IEEE Std 7-4.3.2-2010](#), IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations

[ANSI/IEEE Std 828](#), IEEE Standard for Configuration Management in Systems and Software Engineering

[ANSI/IEEE Std 26514](#), Systems and Software Engineering—Requirements for Designers and Developers of User Documentation

[ANSI/IEEE Std 29148](#), Systems and Software Engineering – Life Cycle Processes – Requirements Engineering

[IEEE Std 344](#), Standard for Seismic Qualification of Equipment for Nuclear Power Generating Stations

[IEEE Std 829](#), IEEE Standard for Software and System Test Documentation

[IEEE Std 830](#), IEEE Recommended Practice for Software Specifications

[IEEE Std 1012](#), IEEE Standard for System and Software Verification

[IEEE Std 1016](#), IEEE Standard for Information Technology – System Design – Software Design Descriptions

[IEEE Std 1016.1](#), IEEE Guide to Software Design Descriptions

[IEEE Std 1028](#), IEEE Standard for Software Reviews and Audits

[IEEE Std 1666](#), IEEE Standard for Standard System C Language Reference Manual

[IEEE Std 12207](#), Systems and Software Engineering – Software Life Cycle Processes

[ISO/IEC/IEEE 24765](#), Systems and Software Engineering – Vocabulary

[ISO/IEC/IEEE 29148](#), Systems and Software Engineering-Life Cycle Processes-Requirements Engineering

[ANSI/ISA S84.00.01](#), Functional Safety: Safety Instrumented Systems for the Process Industry Sector

LANL (internal-only unless noted). Contact the chapter POC for assistance in obtaining LANL documents.

[AP-341-402](#), Engineering Document Management in Operating Facilities

[AP-341-405](#), Identification and Control of Technical Baseline, Variances, Alternate Methods, and Clarifications in Operating Facilities

## 1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

[AP-341-502](#), Management Level Determination and Identification of Quality Assurance and Maintenance Requirements

[AP-341-504](#), Temporary Modification Control

[AP-341-507](#), SSC Control Software Change Package

[AP-341-510](#), Field Walk-down, Data Gathering, and Inspections

[AP-341-519](#), Design Revision Control

[AP-341-605](#), Calculations

[AP-341-601](#), Functions and Requirements Document

[AP-341-602](#), Requirements and Criteria Document (RCD)

[AP-341-611](#), System Design Descriptions

[AP-341-613](#), Instrumentation Set Point Control

[AP-341-616](#), Technical Baseline Change During Design

[AP-341-620](#), Review of LANL Produced Design Documents

[AP-341-621](#), Design Authority Technical Review

[AP-341-622](#), LANL Review of Designs Produced by External Agencies

[AP-341-703](#), Commercial Grade Dedication

[AP-341-801](#), Post Modification/Post Maintenance Testing

[AP-341-802](#), System Health Reporting

[AP-341-901](#), Performing Vital Safety System Assessments

[AP-350-406](#), Startup and Commissioning

[ASM 3400.00.041](#), Goods or Services Requiring Special Review/Approval

[ASM Website](#)

[Conduct of Operations Resources](#)

[EPP-COOP-006](#), LANL COOP [Continuity of Operations] Plan

[ES Division Office \(ES-Div\) Website](#)

[Information System Security Officer \(ISSO\)](#)

[LANL Institutional Supplier List \(IESL\)](#)

[LANL Nuclear Facility List](#)

[LANL Safety Basis Document List](#)

[LA-14167-MS](#), Los Alamos National Laboratory, Concepts of Model Verification and Validation, Thacker et al., 2004

[P204-1](#), Controlled Unclassified Information

[P218](#), Cyber Security Access Controls

[P322-4](#), Laboratory Performance Feedback and Improvement Process

[P330-2](#), Control and Calibration of Measuring and Test Equipment (M&TE)

[P330-3](#), Quality Audits

[P330-5](#), Special Processes

[P330-6](#), Nonconformance Reporting

[P330-8](#), Inspection and Test

[P330-9](#), Suspect/Counterfeit Items (SCI)

[P341](#), Facility Engineering Processes Manual

[P342](#), Engineering Standards

[P343](#), Facility Engineering Training and Qualification Manual

[P805](#), Export Control

[P840-1](#), Quality Assurance for Procurements

[P1020-2](#), Laboratory Document Control

[P1040](#), Software Quality Management

[P1011](#), Computer Hardware, Software, and Services: Exceptions to Acquisition, Maintenance, and Support

[PD110](#), Safety Basis

[PD115](#), LANL Readiness Program

[PD210](#), Cyber Security Program

[PD328](#), LANL Assessment Program

[PD340](#), Conduct of Engineering for Facility Work

[PD370](#), Conduct of Engineering for Research and Development (R&D)

[QPA-DO-FSD-001](#), LANL Safety Software Inventory Procedure

[SBP-15-351](#), Design Basis or Safety Basis Change Review

[SBP111-1](#), Facility Hazard Categorization and Documentation

[SBP112-3](#), Unreviewed Safety Question (USQ) Process

[SBP113-3](#), Unreviewed Safety Issue (USI) Process

[SD210](#), Information Risk Management Framework

[SD330](#), Los Alamos National Laboratory Quality Assurance Program

[STD-342-100](#), Engineering Standards Manual (available externally)

[Merriam-Webster Dictionary](#)

[TeamForge \(LANL-internal\)](#)

[Trac](#)

## ATTACHMENT 1: CHAPTER 21 SUMMARY TABLE

Rev	Date	Description	POC	RM
0	06/23/16	Initial issue	Tobin Oruch, ES-DO	Mel Burnett, CENG-OFF

This table is summary of the entire chapter, addressing both SSC software and Non-SSC software. Every attempt is made to keep it in sync with the requirements in each Section; however:

- Section requirements are also more detailed, and
- In the case of a conflict, contact the Chapter POC (Section requirements will generally supersede those in this summary).

*This search tool may not include the latest files; use individual files for work.*

1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

Table 21.1-ATT1 Chapter 21 Summary by Software Management Level (ML)										
This table is a summary only and does not include all requirement details. See text of Section(s) for details.										
Section Name/ Subsection <sup>1</sup>	Activity Description	Deliverable	SSC Software ML <sup>2</sup>				Non-SSC Software ML <sup>2</sup>			
			1	2	3	4	1	2	3	4
SOFT-GEN, 2	Identify software, evaluate and complete software determination	Software list	R	R	R	G	R	R	R	R
		Form <a href="#">2033</a>	R	R	R	G <sup>3</sup>	R	R	R	G <sup>3</sup>
SOFT-GEN, 3	Plan software-specific activities	Software data sheet form SOFT-GEN-FM03 (SWDS)	R	R	R	-	R	R	R	-
SOFT-GEN, 4	Develop and maintain a software baseline (SWBL)	Software baseline form SOFT-GEN-FM01 (SWBL) <sup>4</sup>	R	R	R	-	R	R	R	-
SOFT-GEN, 4	Control software changes:									
	Less than minor change, minor and major changes made <u>before or after</u> software approval for use (SWAU)	Documented comment in the computer program where feasible	R	R	R	R	R	R	R	R
	Minor or major changes made <u>before</u> SWAU	<a href="#">AP-341-519</a> change documents <sup>5</sup>	R	R	G	G	-	-	-	-
	Minor or major changes made once software is baselined	Revised SWBL <sup>4</sup>	R	R	R	-	R	R	R	-
	Minor or major changes made <u>after</u> SWAU	SSC Software Change Form <a href="#">AP-341-507-FM01</a>	R	R	G	-	-	-	-	-
		Non-SSC Software Change Form SOFT-GEN-FM02 (SWNCP)	-	-	-	-	R	R	G	-
SOFT-GEN, 5	Report problems and take corrective action:									
	Before SWAU	<a href="#">P330-6</a> , <a href="#">P322-4</a> , bug lists, documented comment field in the computer program	R	R	R	G	R	R	R	G
	After SWAU	<a href="#">P330-6</a> , <a href="#">P322-4</a> documents (e.g., Nonconformance Reports [NCRs])	R	R	R	G	R	R	R	G
SOFT-GEN, 6	Manage software (project) risks	Risk list as part of SWDS	R	R	R	G	R	R	R	G

This search tool may not include the latest files; use individual files for work.

1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

Table 21.1-ATT1 Chapter 21 Summary by Software Management Level (ML)										
This table is a summary only and does not include all requirement details. See text of Section(s) for details.										
Section Name/ Subsection <sup>1</sup>	Activity Description	Deliverable	SSC Software ML <sup>2</sup>				Non-SSC Software ML <sup>2</sup>			
			1	2	3	4	1	2	3	4
SOFT-ACQUIRE, 3, 4	Acquire existing LANL software if suitable	Update authorized users/SWDS as required; generate an interface agreement where desired	-	-	-	-	R	R	R	G
SOFT-ACQUIRE, 3, 4	Acquire new software	▪ Statement of Work (SOW)	R	R	R	-	R	R	R	-
		▪ SWDS	R	R	R	-	R	R	R	-
		▪ Commercial Grade Dedication (CGD) Documentation per <a href="#">AP-341-703</a>	R <sup>6</sup>	R <sup>6</sup>	R <sup>6</sup>	-	R <sup>6</sup>	R <sup>6</sup>	R <sup>6</sup>	-
		▪ P-card/Purchase Order (PO)	R	R	R	R <sup>7</sup>	R	R	R	R <sup>7</sup>
		▪ <a href="#">Exhibit H</a>	R	R	R	R <sup>8</sup>	R	R	R	R <sup>8</sup>
SOFT-DESIGN, 3, 6	Develop software design requirements (input)	<ul style="list-style-type: none"> <li>▪ Software design inputs :                             <ul style="list-style-type: none"> <li>○ SSC drawings</li> <li>○ SSC performance specification</li> <li>○ SSC FDD, SDD</li> </ul> </li> <li>▪ Other SSC design inputs/outputs as required</li> </ul>	R	R	G	G	-	-	-	-
		▪ Software system requirements specification (SWYRS)	-	-	-	-	R	R	G	G
SOFT-DESIGN, 4, 7	Design the software (output)	<ul style="list-style-type: none"> <li>▪ Software design outputs :                             <ul style="list-style-type: none"> <li>○ software specification (SWRS)</li> <li>○ software design (SWDD)</li> <li>○ software system hazard analysis and mitigation documentation (SWHA)</li> <li>○ software design traceability documentation (SWTM)</li> <li>○ SOFT-GEN deliverables</li> <li>○ Other software design deliverables as required</li> <li>○ Operations and maintenance (O&amp;M) instructions</li> <li>○ computer model mathematical terms<sup>10</sup></li> </ul> </li> </ul>	R	R	G	G	R	R	G	G

This search tool may not include the latest files; use individual files for work.

1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

**Table 21.1-ATT1 Chapter 21 Summary by Software Management Level (ML)**  
 This table is a summary only and does not include all requirement details. See text of Section(s) for details.

Section Name/ Subsection <sup>1</sup>	Activity Description	Deliverable	SSC Software ML <sup>2</sup>				Non-SSC Software ML <sup>2</sup>			
			1	2	3	4	1	2	3	4
SOFT-DESIGN, 5, 8	Translate the software design into computer program language (code)	<ul style="list-style-type: none"> <li>▪ Computer program language (code):                             <ul style="list-style-type: none"> <li>○ data structures/files,</li> <li>○ source code (where feasible),</li> <li>○ executable code</li> </ul> </li> <li>▪ Computer program listings</li> </ul>	R	R	G	G	R	R	G	G
SOFT-V&V, 4	Review (input) requirements	<ul style="list-style-type: none"> <li>▪ Dispositioned review comments</li> </ul>	R	R	G	G	R	R	G	G
SOFT-V&V, 4	Develop and review the test plan (SWTP)	<ul style="list-style-type: none"> <li>▪ Test plan</li> <li>▪ Dispositioned review comments</li> </ul>	R	R	G	G	R	R	G	G
SOFT-V&V, 4	Review the software design (SWDD)	<ul style="list-style-type: none"> <li>▪ Software design</li> <li>▪ Dispositioned review comments</li> </ul>	R	R	G	G	R	R	G	G
SOFT-V&V, 4	Review computer program code where feasible	<ul style="list-style-type: none"> <li>▪ Dispositioned code review comments</li> </ul>	R	R	-	-	R	R	-	-
SOFT-V&V, 4	Test and review test results	<ul style="list-style-type: none"> <li>▪ Interim tests and test (qualification) reports</li> <li>▪ Simulated acceptance test and acceptance test report</li> <li>▪ Acceptance test and test (qualification) report (SWTR)</li> <li>▪ Dispositioned review comments</li> </ul>	R	R	G	G	R	R	G	G
SOFT-V&V, 4	Review and approve software for use (SWAU)	<ul style="list-style-type: none"> <li>▪ SWAU request package</li> <li>▪ Dispositioned review comments</li> <li>▪ Software approval for use form SOFT-V&amp;V-FM01, SWAU</li> </ul>	R	R	G	-	R	R	G	-
SOFT-INV, 3, 4	Inventory safety software	<ul style="list-style-type: none"> <li>▪ Updated institutional-level safety software inventory</li> </ul>	R	R	R <sup>11</sup>	-	R	R	R <sup>11</sup>	-
		<ul style="list-style-type: none"> <li>▪ ES-Div: Updated ES Non-SSC software inventory</li> </ul>	-	-	-	-	R	R	R <sup>11</sup>	-
SOFT-INV, 3, 4	Inventory non-safety software	<ul style="list-style-type: none"> <li>▪ ES-Div: Updated ES non-SSC software inventory</li> <li>▪ ES-Div: Form <a href="#">2033</a> to ESSC</li> </ul>	-	-	-	-	R	R	R	R

This search tool may not include the latest files; use individual files for work.

1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

Table 21.1-ATT1 Chapter 21 Summary by Software Management Level (ML)										
This table is a summary only and does not include all requirement details. See text of Section(s) for details.										
Section Name/ Subsection <sup>1</sup>	Activity Description	Deliverable	SSC Software ML <sup>2</sup>				Non-SSC Software ML <sup>2</sup>			
			1	2	3	4	1	2	3	4
SOFT-MAINT, 3, 4	Use and maintain the software	<ul style="list-style-type: none"> <li>▪ Operation and maintenance (O&amp;M) documentation</li> <li>▪ Operational event documentation</li> <li>▪ Application logs</li> <li>▪ Licenses and registrations</li> <li>▪ Access controls</li> <li>▪ Computer system vulnerability protections</li> <li>▪ Problem reports and corrective actions per SOFT-GEN</li> <li>▪ In-use test plans, tests, review documentation, and test reports per SOFT-V&amp;V</li> <li>▪ Updated, configuration management, SWDS and deliverables as described on SWDS and SWBL per SOFT-GEN</li> <li>▪ Accurate software inventory per SOFT-INV</li> <li>▪ Other quality assurance control documentation (e.g., training records, assessments)</li> </ul>	R	R	G	G	R	R	G	G
SOFT-MAINT, 3, 4	Retire the software	<ul style="list-style-type: none"> <li>▪ Software removed to prevent unintended or routine use</li> <li>▪ Terminated licenses/agreements</li> <li>▪ Updated software inventory(ies)</li> <li>▪ Cancelled or revised affected documents</li> <li>▪ Software record</li> </ul>	R	R	R	R	R	R	R	R

This search tool may not include the most files; use individual files for work.



1 Section SOFT-GEN: General Software Requirements

Rev. 0, 06/23/16

**Table 21.1-ATT1 Chapter 21 Summary by Software Management Level (ML)**  
 This table is a summary only and does not include all requirement details. See text of Section(s) for details.

Section Name/ Subsection <sup>1</sup>	Activity Description	Deliverable	SSC Software ML <sup>2</sup>				Non-SSC Software ML <sup>2</sup>			
			1	2	3	4	1	2	3	4

Notes:

- <sup>1</sup> ESM Chapter 21 section name and subsection number.
- <sup>2</sup> See Form 2033 for a crosswalk between MLs and software type, category and SRL. R = Required. “-” = Required but graded. “-” = Not required.
- <sup>3</sup> Complete Form 2033 for ML-4 software only if there is a reasonable probability that it could be ML-3, ML-2, or ML-3 software, or, the highest function of the SSC is greater than ML-4.
- <sup>4</sup> See SOFT-GEN-FM01, *Software Baseline Form (SWBL)*; the SWBL may be included as part of a system design description (SDD) or other controlled document. The SWBL must include documents and computer program files.
- <sup>5</sup> Includes Field Change Notices (FCNs), Field Change Requests (FCRs), Interim Work Requests (IWRs), Design Revision Notices (DRNs), Design Change Forms (DCFs) as described in AP-341-519 unless specifically superseded by another administrative procedure.
- <sup>6</sup> Required only for safety software only that is not procured from an NQA-1 qualified supplier.
- <sup>7</sup> A PO or a Purchase Card (P-card) can be used for ML-4 software; a P-card cannot be used for ML1 through ML-3 software.
- <sup>8</sup> Required if the supplier comes onsite (at LANL) to install and/or maintain software.
- <sup>9</sup> Required if software is acquired from LANL organizations, DOE sites or government agencies.
- <sup>10</sup> Required for computer program models only.
- <sup>11</sup> Required if safety software.

This search tool may not include the latest files; use individual files for work.



SWBL No.:

Rev.:

**1.0 ASSOCIATED SOFTWARE INFORMATION**

1.1 Software Name:				
1.2 TA No.:		1.3 Facility No.:		1.4 Facility Name:
1.5 SWID No.:		1.6 Equipment ID:		
1.7 Software Description:				

**2.0 SOFTWARE BASELINE – DOCUMENTS**

2.1 Document Number (including revision)	2.2 Document Title	2.3 Document Description
2.4 Document files are located in:		

**3.0 SOFTWARE BASELINE – COMPUTER PROGRAMS**

3.1 Computer Program File/Folder Name (including revision)	3.2 Computer Program Description
3.3 Computer program files are located in:	

*This search tool may not include the latest files; use individual files for work.*

**LANL**



# Conduct of Engineering Software Baseline Form (SWBL) Example For CBT Software

SWBL No.: | SWBL-03-2332-12345

Rev.: | 0

Page 6 of 6

4.0 PREPARED BY/REVIEWED BY/APPROVED BY	
4.1 Prepared By – Software Owner (SO), (Name, Z number, Organization, Signature and Date):	
Tom Thompson, 143987, NEN-2, 10/1/2015	
4.2 Reviewed and Approved By – Software Responsible Line Manager (SRLM), (Name, Z number, Organization, Signature and Date):	
Michael Bradley, 114987, NEN-2, 10/2/2015	
4.3 Reviewed and Approved By – Others as required by SRLM, (Name, Z number, Organization, Signature and Date):	
None Required by SRLM (Not Applicable)	

5.0 REVISIONS		
5.1 Revision	5.2 Date	5.3 Change Description and Reason for Change
0	10/02/2015	Original Issue

## LANL

STD-342-100, Chapter 21, Software, Section SOFT-GEN: General Software Requirements  
SOFT-GEN-FM01, Software Baseline Form (SWBL) with Example, Rev. 0 (06/23/16)

*This search tool may not include the latest files; use individual files for work.*



# Conduct of Engineering Software Baseline Form (SWBL) Instructions with Example

## GENERAL

A software baseline (SWBL) is a specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for use and further development, and that can be changed only by using an approved change control process. (Ref. NQA-1). A software baseline includes the computer program (code and [configuration] data) and the computer program documentation. In layman's terms, one may think of this as the information and computer program files that are needed to run the software for a specific application. See attached SWBL example for items that may be listed as part of a SWBL.

For additional detail on developing a software baseline (SWBL), see STD-342-100, *Engineering Standards Manual, Chapter 21, SOFT-GEN*.

LANL personnel: Endeavor to use Chapter forms as-is and report issues and improvement ideas to the Chapter 21 POC. POC may authorize other methods equivalent to chapter forms in writing.

LANL subcontractors must use Chapter forms to satisfy Chapter requirements for SSC software. For Non-SSC software, subs may either use their own forms or integrate, adapt, and reformat the forms; either approach is acceptable so long as key functions, data, and approvals are retained.

## HEADER

Field	Entry Information
Software Name	Enter the name of the software in the upper right hand corner of the header as shown in the example form (e.g. Software Baseline for CBT).
SWBL No.	The numbering scheme of this form follows that of the SRLM's document control/records management system. For software used in the operation of existing or new LANL facilities, use numbering scheme and process specified in <a href="#">AP-341-402</a> , <i>Engineering Document Management in Operating Facilities</i> , e.g., SWBL-03-0216-1234.
Rev.	Enter the revision number. Initial approved issue of the software baseline is 0.

## 1.0 ASSOCIATED FACILITY, SYSTEM AND SOFTWARE INFORMATION

Field	Entry Information
1.1	Enter the name of the software.
1.2	Enter the two-digit Technical Area (TA) number(s) where the software is used.
1.3	Enter the facility number(s) where the software is used.

## LANL

Field	Entry Information
1.4	Enter the facility name(s) where the software is used.
1.5	For ES- and SB-Div Non-SSC software and SSC software, enter the software identification number (SWID). The SWID should be the same SWID as on the Form 2033. SWIDs are obtained in accordance with AP-341-402, <i>Engineering Document Management in Operating Facilities</i> . Ensure the SWID is part of the SWBL and subsequent software documentation.
1.6	For SSC software, enter the equipment identification (CLI) number associated with the software per ESM Chapter 1, Section 200 (e.g., 03-0216-HVAC-BAS-1). For Non-SSC software, enter NA.
1.7	Enter a description of the software. Include the primary functions and systems associated with the software.

## 2.0 SOFTWARE BASELINE - DOCUMENTS

Field	Entry Information
2.1	Enter the document number in accordance with the SRLM's document control records/management system. Include revision number.
2.2	Enter the document title.
2.3	Enter a description of the document.
2.4	Enter the Attachment name and physically attach electronic copies of the document files (e.g. on a compact disc or other media). Ensure the file names match the names listed in the SWBL.  <b>Note:</b> Methods other than physically attaching the files to the SWBL (e.g., hyperlinking to documents in a electronic data management system (EDMS) may be used as long as they conform to the governing document control/records management program and ensure file integrity and retrievability.

## 3.0 SOFTWARE BASELINE – COMPUTER PROGRAMS

Field	Entry Information
3.1	Enter the computer program name, revision and where possible, file extension (e.g. .exe). Use of a file folder that includes several files is acceptable for programs where multiple files make listing them separately too onerous. If using unique file folder, enter "folder" in this box after the name and revision.
3.2	Enter the computer program description. Describe what it does/how it supports the overall use of the software.
3.3	Enter the Attachment name and physical attach electronic copies of the computer program files (e.g., on a compact disc or other media). Ensure the file names match the names listed in the SWBL.  <b>Note:</b> Methods other than physically attaching the files to the SWBL may be used as long as they conform to the governing document control/records management

## LANL

Field	Entry Information
	program and ensure file integrity and retrievability.

#### 4.0 PREPARED BY/REVIEWED BY/APPROVED BY

Field	Entry Information
4.1	Enter SO name, Z number, organization, signature, and date. For assistance in determining the SO, see SOFT-GEN Appendix C: <i>SO and SRLM Decision Diagram for FAC-COE</i> .
4.2	Review and as appropriate, approve by entering SRLM name, Z number, organization, signature, and date. For assistance in determining the SRLM, see SOFT-GEN Appendix C: <i>SO and SRLM Decision Diagram for FAC-COE</i> .
4.3, 4.n+1...	Add additional reviewers/approvers as required by the SRLM. Review and as appropriate, approve by entering additional reviewer/approver name, Z number, organization, signature, and date.

#### 5.0 REVISIONS

Field	Entry Information
5.1	Enter the revision number here and update the revision number in the form header.
5.2	Enter the date of the revision.
5.3	Enter a summary description of what was changed and the reason for the change.

This search tool may not include the latest files; use individual files for work.

## LANL

### 1.0 ASSOCIATED SOFTWARE INFORMATION

1.1 Software Name		Combustible Material Tracker (CBT) Software			
1.2 TA No.:	03	1.3 Facility No.:	2333	1.4 Facility Name:	Nonproliferation Security Facility (NSF)
1.5 SWID:	03-2333-0047	1.6 Equipment ID:	NA		
1.7 Software Description:					
The software is used as a tool to track combustible material.					

### 2.0 SOFTWARE BASELINE – DOCUMENTS

2.1 Document Number (including revision)	2.2 Document Title	2.3 Document Description
SWDS-15-03-NSF-0001	CBT Software Data Sheet (SWDS)	Documents software-specific information to supplement ESM Chapter 21
SWRS-15-03-NSF-0002	CBT Software Requirements Specification	Requirements document
SWTM-15-03-NSF-0003	CBT Software Requirements Traceability Matrix	Traceability document that traces requirements from the SRS through the software lifecycle.
SWRL-15-03-NSF-0004	CBT Software Risk List (Register)	Listing and ranking of risks for the software throughout its lifecycle.
SWHA-15-03-NSF-0005	CBT Software Hazard Analysis	Hazard analysis of software and integrated software/hardware system.
SWDD-15-03-NSF-0006	CBT Software Design Document	Software design. Includes system configuration including associated hardware and facility interfaces and key support software configuration
SWTP-15-03-NSF-0007	CBT Software Acceptance Test and In-Use Test Plan	Test plan document that describes acceptance and in-use tests.
SWTR-15-03-NSF-0008	CBT Software Acceptance Test Report	Describes acceptance test results and acceptance.
SWMM-15-03-NSF-0009	CBT Software Maintenance Manual	Detailed technical manual for software subject matter expert (SME) to use for troubleshooting and maintaining the software.

## LANL



Conduct of Engineering  
**Software Baseline Form (SWBL) Example For CBT Software**

SWBL No.: | SWBL-03-2332-12345

Rev.: | 0

Page 5 of 6

**2.0 SOFTWARE BASELINE – DOCUMENTS**

2.1 Document Number (including revision)	2.2 Document Title	2.3 Document Description
SWUM-15-03-NSF-0010	CBT Software User Manual	User manual. Includes general instructions for use, including limitations.
AP-15-03-NSF-0007	CBT Software Administrative Procedure	Administrative procedure that integrates the software into the facility day-to-day operations.
SWLI-15-03-SNF-1110	CBT Software License/License Key	License and license key required to use the software
2.4 Document files are located in:		
<a href="#">Attachment A: CBT SWBL-15-03-2332-0234 Document Files.</a>		

**3.0 SOFTWARE BASELINE – COMPUTER PROGRAMS**

3.1 Computer Program File/Folder Name (including revision)	3.2 Computer Program Description
CBT 1.02.exe	Primary executable computer program that provides overall system function
CBT Master Database 1.2. mdb	The primary database of material.
Microsoft Windows XP Professional SP1.1	Off the shelf (OTS) support software that is the operating system on client computers.
Microsoft Server 2003 R2 Folder	OTS support software that is the operating system on computer servers.
Microsoft Access Database 2000 SP1.01 Folder	OTS support software that provides data storage on servers and client computers.
Microsoft Outlook 2013 SP1.0 Folder	OTS support software that provides email notifications of transactions.
3.4 Computer program files are located in:	
<a href="#">Attachment B: CBT SWBL-15-03-2332-0234 Computer Program Files.</a>	

**LANL**

STD-342-100, Chapter 21, Software, Section SOFT-GEN: General Software Requirements  
 SOFT-GEN-FM01, Software Baseline Form (SWBL) with Example, Rev. 0 (06/23/16)

This search tool may not include the latest files; use individual files for work.





SWBL No.:

Rev.:

4.0 PREPARED BY/REVIEWED BY/APPROVED BY	
4.1 Prepared By – Software Owner (SO), (Name, Z number, Organization, Signature and Date):	
4.2 Reviewed and Approved By – Software Responsible Line Manager (SRLM), (Name, Z number, Organization, Signature and Date):	
4.3 Reviewed and Approved By – Others as required by SRLM, (Name, Z number, Organization, Signature and Date):	

5.0 REVISIONS		
5.1 Revision	5.2 Date	5.3 Change Description and Reason for Change

*This search tool may not include the latest files; use individual files for work.*

**LANL**



**Conduct of Engineering  
Non-SSC Software Change Package Form  
(SWNCP)**

SWNCP No.:

Rev.:

Page 1 of 3

**1.0 SOFTWARE CHANGE REQUEST INFORMATION**

1.1 Software Name:	1.2 SWID No.:	1.3 Request Date:
1.4 Software Change Request Title:		
1.5 Software Type:		1.6 Management Level (ML) Application:
<input type="checkbox"/> Safety <input type="checkbox"/> Non-Safety	<input type="checkbox"/> ML-1 <input type="checkbox"/> ML-2 <input type="checkbox"/> ML-3	<input type="checkbox"/> ML-4
1.7 TA No.:	1.8 Facility No.:	1.9 Facility Name:
1.10 Change Type:		Major: <input type="checkbox"/> Minor: <input type="checkbox"/>
1.11 Type of Reason for Change:		
Defect: <input type="checkbox"/> New and/or Modified Requirement: <input type="checkbox"/>		
1.12 Reason for Change:		
1.13 Change Description:		
1.14 Requestor: (Name, Z Number if applicable, Organization)		
1.15 Software Change Impact Description:		
1.16 Affected Software Baseline - Documents		
Document No. (including revision)	Document Title	Document Description
1.17 Affected Software Baseline – Computer Programs		
Computer Program File Name (Including revision)	Computer Program Description	
1.18 Planned Type of Verification and Validation (V&V) for the Change		
Review(s): <input type="checkbox"/> Test(s): <input type="checkbox"/> Alternate Calculation(s): <input type="checkbox"/>		
1.19 V&V Description:		

This search tool may not include the latest files; use individual files for work.

**LANL**



## Conduct of Engineering Non-SSC Software Change Package Form (SWNCP)

SWNCP No.:

Rev.:

Page 2 of 3

<p>1.20 Is this software used for the administrative control in a hazard category 2 or 3 nuclear facility, high hazard nonnuclear facility, moderate hazard nonnuclear facility, and/or accelerator facility?</p> <p>If "No" is checked, go to Step 2.1. If "Yes" is checked, go to Step 1.21.</p>	<p>Yes <input type="checkbox"/> No <input type="checkbox"/></p>
<p>1.21 Does this software change require engagement of the evaluation processes of SBP-15-351, <i>Design Basis or Safety Basis Change Review</i> or other safety basis (SB) reviews?</p> <p>If "No" is checked, go to Step 2.1. If "Yes" is checked, complete safety basis evaluation(s), before approval for use (Section 4.) and go to Step 1.22. Consider completing evaluations before proceeding further.</p>	<p>Yes <input type="checkbox"/> No <input type="checkbox"/></p>
<p>1.22 Design basis and/or safety basis evaluation(s) document number(s):</p>	
<b>2.0 SOFTWARE CHANGE REQUEST REVIEW AND ACCEPTANCE</b>	
<p>2.1 Software Responsible Line Manager (SRLM) Review and Acceptance</p> <p style="text-align: right;"> <input type="checkbox"/> Accepted                <input type="checkbox"/> Accepted with Comments                <input type="checkbox"/> Rejected         </p> <p>If "Accepted with Comments" or "Rejected" is checked, then provide comments or justification below:</p>	
<p>2.2 SRLM: (Name, Z Number if applicable, Organization, Signature, and Date)</p>	
<p>2.3 FDAR Review and Acceptance (required only for major changes of software used for administrative control in a LANL facility)</p> <p style="text-align: right;"> <input type="checkbox"/> NA                <input type="checkbox"/> Accepted                <input type="checkbox"/> Accepted with comments                <input type="checkbox"/> Rejected         </p> <p>If "Accepted with Comments" or "Rejected" is checked, then provide comments or justification below:</p>	
<p>2.4 FDAR: (Name, Z Number, Organization, Signature, and Date)</p>	

<b>3.0 SOFTWARE CHANGE VERIFICATION AND VALIDATION (V&amp;V) &amp; RECOMMENDATION</b>		
<p>3.1 The software change has been V&amp;V'd in accordance with the attached V&amp;V planning documentation. The V&amp;V results and the revised software baseline (without SRLM signature) are attached.</p>	<p>Recommended for Approval for Use <input type="checkbox"/></p>	<p>Not Recommended for Approval for Use <input type="checkbox"/></p>
<p>If "Not Recommended for Approval for Use", then provide reason(s) for disapproval below:</p>		
<p>SO: (Name, Z Number if applicable, Organization, Signature, and Date)</p>		

### LANL



# Conduct of Engineering Non-SSC Software Change Package Form (SWNCP)

SWNCP No.:

Rev.:

Page 3 of 3

## 4.0 SOFTWARE CHANGE APPROVAL FOR USE

4.1 Software Responsible Line Manager (SRLM) Review and Acceptance

Approved for Use       Rejected

If "Rejected" is checked, then provide reason below:

4.2 SRLM: (Name, Z Number if applicable, Organization, Signature, and Date)

4.3 FDAR Review and Acceptance (for major changes only)

Not Applicable (NA)       Approved for Use       Rejected

If "Rejected" is checked, then provide reason below:

4.4 FDAR: (Name, Z Number, Organization, Signature, and Date)

## 5.0 ATTACHMENTS

5.1 Attachment No.	5.2 Attachment Date	5.3 Attachment Title

## 6.0 REVISIONS

6.1 Revision	6.2 Date	6.3 Change Description and Reason for Change

This search tool may not include the latest files; use individual files for work.

### LANL



# Conduct of Engineering Non-SSC Software Change Package Form (SWNCP) Instructions

Page 1 of 7

## GENERAL

These instructions are for completing the form and are not part of the completed form.

LANL personnel: Endeavor to use Chapter forms as-is and report issues and improvement ideas to the Chapter 21 POC. POC may authorize other methods equivalent to chapter forms in writing.

LANL subcontractors must use Chapter forms to satisfy Chapter requirements for SSC software. For Non-SSC software, subs may either use their own forms or integrate, adapt, and reformat the forms; either approach is acceptable so long as key functions, data, and approvals are retained.

This form does not apply to SSC Software. For SSC software, see AP-341-507, *SSC Software Control Software Change Package* or AP-341-519, *Design Revision Control*.

This form does not apply to less than minor computer program changes as defined below.

**Less Than Minor Computer Program Change.** A change or bug fix that is not a major or minor computer program change but:

- adds, deleted and/or modifies ML-4 performance function code,
- adds, deletes and/or modifies code that does not modify ML-1, ML-2, ML-3 performance function, or
- imparts changes without adding, deleting or modifying design and/or analysis output values (all MLs).

Examples: Modify code to increase the ramp time on an ML-4 softstart pump. Install security patch/service pack updates. An OTS software patch that includes a code change to prevent a screen from "freezing" or loading slowly (all MLs). Add/modify code clarifying notes (all MLs). Modify code to produce multiple reporting formats (all MLs).

For less than minor computer program changes, ensure they are made by a competent individual knowledgeable in the software and add a brief description, name of who is making, and the date that the less than minor change is made in the computer program code at the time the change is made.

**Note:** Upon completion of changes, the SRLM is responsible for:

- (a) incorporating changes into the software baseline as soon as practical after changes are approved, and
- (b) retaining Non-SSC Software Change packages as a record in accordance with P1020-1, *Laboratory Records Management* or the governing records management process.

## LANL

STD-342-100, Chapter 21, Software, Section SOFT-GEN: General Software Requirements, SOFT-GEN-FM02, Non-SSC Software Change Package Form Instructions (SWNCP), Rev. 0 (06/23/16)

**HEADER (COMPLETED BY REQUESTOR)**

Field	Entry Information
SWNCP No.	Number this form in accordance with the governing document control procedure. For LANL owned Non-SSC software in the scope of the Facility Conduct of Engineering (FAC-COE), use the numbering scheme and process specified in <a href="#">AP-341-402</a> , <i>Engineering Document Management in Operating Facilities</i> .
Rev.	Enter "0" for initial revision and 1, 2, etc. for subsequent revisions.

**1. SOFTWARE CHANGE REQUEST INFORMATION (COMPLETED BY REQUESTOR)**

Field	Entry Information
1.1	Enter the name of the software to be changed.
1.2	Enter the software identification number (SWID). The SWID should be the same SWID as on the Form <a href="#">2033</a> . SWIDs are obtained in accordance with <a href="#">AP-341-402</a> , <i>Engineering Document Management in Operating Facilities</i> .
1.3	Enter the date the Non SSC software change request was initiated.
1.4	Enter a specific and succinct Non-SSC software change request title.
1.5	Indicate the software type to be changed by checking either the safety or non-safety box. If the software is used for both safety and non-safety applications, check the safety box only.
1.6	Indicate the management level (ML) application by checking one of the boxes. If used for multiple ML applications check the highest (lowest numbered) box. (e.g., for software used in ML-2 and ML-3 applications, check the ML-2 box.)
1.7	Enter the TA number of the technical area associated with the software change. If the software is used at various LANL TAs, enter "multiple".
1.8	Enter the facility number of the facility associated with the software change. If the software is used at various LANL facilities, enter "multiple".
1.9	Enter facility name of the facility associated with the software change. If the software is used at various LANL facilities, enter "multiple".
1.10	<p>Indicate whether the change is a major or minor change by checking either the major or minor box. Note that a change can include a change to the computer program (including data) or changes to documentation. Use the following definitions.</p> <p><b>Major Computer Program Change:</b> A change or bug fix that:</p> <ul style="list-style-type: none"> <li>▪ the Software Responsible Line Manager (SRLM) or computer program supplier designates as a major change,</li> <li>▪ adds or deletes an ML-1, ML-2 or ML-3 SSC performance function,</li> <li>▪ modifies ML-1 or ML-2 SSC performance function code, excluding clarifying notes,</li> </ul>

**LANL**

Field	Entry Information
	<ul style="list-style-type: none"> <li>▪ adds, deletes or modifies design and/or analysis output values of ML-1, ML-2 or ML-3 calculations,</li> <li>▪ recodes to another language, or</li> <li>▪ modifies a significant number of lines of code.</li> </ul> <p><b>Note:</b> A Major OTS computer program change is often indicated with increment increase in version number (e.g., change from version 1 to 2). An evaluation of the software however, is required to determine whether the version release is a major change.</p> <p><u>Examples:</u> A change from Delta V control system software from version 7.0 to version 8.0. A change that adds code to implement an interlock functional performance requirement that an ML-3 laser system cannot be activated until area doors are locked. A change that modifies code on ML-2 ventilation system backdraft damper so that damper closure does not slam shut and potentially damage the damper assembly. A change in the algorithm or databased used for calculating the water flow rate in an ML-3 fire protection piping system design. A change in coding language from C to C++. A version change where 40% of the lines of code were modified.</p> <p><b>Minor Computer Program Change:</b> A change or bug fix that is not a major computer program change but:</p> <ul style="list-style-type: none"> <li>▪ adds or deletes an ML-4 SSC performance function,</li> <li>▪ modifies ML-3 SSC performance function code, excluding clarifying notes, or</li> <li>▪ adds, deletes or modifies design and/or analysis output values of ML-4 calculations</li> </ul> <p><b>Note:</b> A minor OTS software change is often indicated with a fractional increase in version number (e.g., 1.1 or 1.01). An evaluation of the software however, is required to determine whether the version release is a minor change.</p> <p><u>Examples:</u> Add code to implement automatic pump shut-off performance requirement on ML-4 sump low-level alarm. Modify code to fix a coding error on an ML-3 heating/cooling system so that cooling, rather than heating activates at high temperatures. Change the algorithm for calculating the current that flows in an ML-4 electric power system under abnormal conditions.</p> <p><b>Minor Document Change.</b> A document change, as defined by the governing document control program, that includes but is not limited to inconsequential editorial corrections, grammatical and spelling changes, organizational name and acronym changes, and similar type changes. A minor document change <u>does not</u> include revisions, changes, or modifications to a document (e.g., procedure, work instruction, drawing, etc.) which impact the effective implementation of the requirement(s). (Ref. P1020-2).</p> <p><b>Major Document Change.</b> A document change that is not a minor document change.</p>
1.11	<p>Enter the type of reason for the change request. Check all that apply.</p> <p><b>Defect:</b> 1. a problem which, if not corrected, could cause an application to either fail or to</p>

Field	Entry Information
	<p>produce incorrect results. 2. an imperfection or deficiency in a project component where that component does not meet its requirements or specifications and needs to be either repaired or replaced. (Ref. ISO/IEC/IEEE 24765:2010-12-15, Systems and Software Engineering – Vocabulary).</p> <p><b>Requirement Change:</b> New and/or modified requirements.</p>
1.12	<p>Describe the reason (rationale) for the change request. Describe “why” the change is needed. Enter the Nonconformance Report (NCR) name and number or other problem/error report that may apply. Indicate what could happen if the change is not implemented.</p>
1.13	<p>Describe “what” the change is to sufficient detail to allow a reviewer to understand, review and as appropriate, approve the change request.</p>
1.14	<p>Enter the name, Z number if applicable and organization of the person requesting the change (Requestor). This should typically be the SO although others may also be Requestors. For assistance in determining the SO, see SOFT-GEN Appendix C: <i>SO and SRLM Decision Diagram for FAC-COE</i>.</p>
1.15	<p>Describe the impacts of the software change. Identify potential risks associated with the change.</p>
1.16	<p>List affected software baseline documents. Provide document number, revision, document title, and document description. Integrate software design requirements, as applicable with the SSC technical baseline documents. See STD-342-100, <i>Engineering Standards Manual, Chapter 1, Section Z10, General</i>; AP-341-616, <i>Technical Baseline Change During Design</i>; and AP-341-405, <i>Identification and Control of Technical Baseline, Variances, Alternate Methods, and Clarifications in Operating Facilities</i>.</p>
1.17	<p>List affected software baselines computer programs. Provide computer program filename, revision, and computer program description.</p>
1.18	<p>Indicate the planned type of verification and validation (V&amp;V) for the change by checking the review(s), test(s) and/or alternate calculations(s) box. Check all that apply.</p>
1.19	<p>Describe the V&amp;V to sufficient detail to allow a reviewer to understand, review and as appropriate, approve the change request and the planned approach to V&amp;V. Describe the approach to retesting and acceptance of test results.</p>
1.20	<p>Indicate whether the software is used for the administrative control in a hazard category 2 before 3 nuclear facility, high hazard nonnuclear facility, moderate hazard nonnuclear facility, and/or accelerator facility by checking either the “Yes” or “No” box. If “No” is checked, go to Step 2.1. If “Yes” is checked, go to Step 1.21.</p>



Field	Entry Information
1.21	<p>Indicate if a safety basis (SB) evaluation of the change is required per the applicable SB procedures listed on <a href="#">SB website</a> by checking either the yes or no box. (e.g., <a href="#">SBP-16-351</a>, <i>Design Basis or Safety Basis Change Review</i>, <a href="#">SBP113-3</a>, <i>Unreviewed Safety Issue Process</i>.)</p> <p>If “Yes” is checked, then the SO ensures the required actions are taken, attaches evidence of completion prior to submitting the SWNCP for approval for use (Section 4.0). If “No” is checked, proceed to Section 2.0. As required, contact an FDAR or <a href="#">a safety basis representative</a> for assistance.</p>
1.22	<p>Enter the attached design basis and/or safety basis document numbers associated with the evaluation(s) (e.g., per SBP-112-3, USQD TA55-16-0234-D).</p>

**2. REVIEW AND ACCEPTANCE (COMPLETED BY SRLM AND AS REQUIRED FDAR OR SUBCONTRACTOR TECHNICALLY QUALIFIED PERSON)**

Field	Entry Information
2.1	<p>The SRLM evaluates the change request and checks the appropriate decision box. Describe comments when “Approved with comments” box is checked. Provide reason(s) for rejection when “Rejected” box is checked.</p> <p><b>Note:</b> “Accepted” means that the requested change is acceptable as proposed. “Accepted with Comments” means that the change is authorized as described in comments.</p>
2.2	<p>Enter the SRLM name, Z number if applicable, organization, signature, and date.</p>
2.3	<p>If the change is a major change for Non-SSC software that is used for administrative control in a LANL facility, then the FDAR evaluates the change request and checks the appropriate decision box.</p> <p><b>Note:</b> For software used among multiple facilities at LANL, where there are multiple FDARs, the LANL Facility Design Authority, or a representative sampling of FDARS (but no fewer than two) must review the request.</p> <p>If the software change is a minor change or a change to software that is not used for administrative control in a LANL facility, then the SRLM checks the “NA” decision box and proceed to Step 2.5.</p>
2.4	<p>Enter the FDAR’s name, Z number, organization, signature, and date for all decisions other than “NA” in 2.3. If “NA” in 2.3, leave blank.</p>

**LANL**

**3.0 VERIFICATION AND VALIDATION (V&V) (COMPLETED BY SO)**

Field	Entry Information
3.1	<p>Plan and complete V&amp;V, including retesting as required and acceptance of test results. Control changes. Attach V&amp;V plan documentation, V&amp;V results documentation, and revised software baseline (SWBL) completed with the exception of the SRLM signature. For software changes used for ML-1, ML-2 or ML-3 administrative control in a LANL facility, also include the planned implementation sequence.</p> <p>Base the extent of V&amp;V on the complexity of the software, the degree of standardization, the similarity with previously approved software, and the importance to safety. Major changes should be acceptance tested per SOFT-V&amp;V. For minor changes, if review alone is inadequate to determine if requirements are satisfied, use alternate calculations and/or develop and integrate (interim) tests to support the review.</p> <p>For SSC or Non-SSC computer programs that were originally commercially dedicated: CGD major changes. CGD minor changes if the change affects the critical characteristic(s) that pertain to the functionality of the computer program and as applicable, the associated SSC.</p> <p>Evaluate and indicate whether the change is recommended/not recommended "for approval for use" or not by checking the appropriate box and entering SO name, Z number if applicable, organization, signature and date of recommendation. For assistance in determining the SO, see SOFT-GEN Appendix C: <i>SO and SRLM Decision Diagram for FAC-COE</i>.</p> <p>If recommended for approval for use, proceed to Step 4.1. If not recommended for approval for use, do not proceed to Step 4.1. Provide reason for recommending disapproval and retain the form as a record in accordance with P1020-1, <i>Laboratory Records Management</i> (or for subcontractors, the subcontractors governing records management process).</p>

**4.0 APPROVAL FOR USE (COMPLETED BY SRLM AND AS REQUIRED, FDAR)**

Field	Entry Information
4.1	The SRLM evaluates the change request for conformance to requirements and checks the appropriate decision box. Provide reason(s) for rejection when "Rejected" box is checked. For assistance in determining the SRLM, see SOFT-GEN Appendix C: <i>SO and SRLM Decision Diagram for FAC-COE</i> .
4.2	Enter the SRLM name, Z number if applicable, organization, signature, and date on the SWNCP and the attached revised SWBL.
4.3	If the change is a major change for Non-SSC software used for control in a LANL facility, then the FDAR evaluates the change and implementation sequence of the change in accordance with AP-341-621, <i>Design Authority Technical Review</i> and checks the appropriate decision box.

**LANL**

Field	Entry Information
	<p><b>Note:</b> For software used among multiple facilities at LANL, where there may be multiple FDARs, the LANL Facility Design Authority, or a representative sampling of FDARs (but no fewer than two) must review the request.</p> <p>Provide reason(s) for rejection when “Rejected” box is checked.</p> <p>If the software change is a minor change or, a major change for software that is not used for administrative control in a LANL facility, then the SRLM checks the “NA” decision box.</p>
4.4	<p>Enter the FDAR name, Z number, organization, signature, and date for all decisions other than “NA” in 4.3. If “NA” in 4.3 is checked, leave blank. The SWNCP is complete.</p> <p>The SRLM retains the SWNCP as a record in accordance with P1020-1, <i>Laboratory Records Management</i> or the governing records management process.</p>

## 5.0 ATTACHMENTS

Field	Entry Information
5.1	Provide attachments as appropriate. Attachments are not mandatory. Enter the attachment number.
5.2	Enter the attachment date.
5.3	Enter the attachment title, including revision as appropriate.

## 6.0 REVISIONS

Field	Entry Information
6.1	If the SWNCP requires revision, then revise in accordance with the governing document control program. Enter the revision number: 1, 2, etc. as appropriate.
6.2	Enter the revision approval date.
6.3	Describe the change and why the change was made.

## LANL

SWDS No.:

Rev.:

**1.0 GENERAL INFORMATION**

1.1 Software Name:					
1.2 TA Number(s):		1.3 Facility No.(s):		1.4 Facility Name(s):	
1.5 SWID No.:			1.6 Equipment ID:		
1.7 Software Summary Description/Need:					
1.8 SSC Software/Non-SSC Software:		SSC <input type="checkbox"/> Non-SSC <input type="checkbox"/>			
1.9 Highest ML Associated with the Software		ML-1 <input type="checkbox"/> ML-2 <input type="checkbox"/> ML-3 <input type="checkbox"/> ML-4 <input type="checkbox"/>			
1.10 Software Type		Safety Software <input type="checkbox"/> Non-Safety Software <input type="checkbox"/>			

**2.0 ROLES AND DATA SHEET APPROVALS**

2.1 Name (Last, First):	2.2. Z Number:	2.3 Data Sheet Approval and Approval Date	2.4 Role:													
			S	SRLM	SD	SD RLM	SU	SU RLM	FDAR	DA	Other	Other	Other			

2.5 Role Comments/Clarifications:

**LANL**

SWDS No.:

Rev.:

3.0 SOFTWARE PROJECT MANAGEMENT					
3.1 Estimated Rough Order of Magnitude (ROM) Software Project Management Cost, K\$					
Fiscal Year (FY):					Total, K\$ (FY16 -20):
2016	2017	2018	2019	2020	
4.0 REQUIREMENTS					
4.1 Software Requirements and Hierarchy:					
4.2 Key Planning Assumptions/Constraints:					
5. DELIVERABLES					
5.1 Deliverables:					
6. ACQUISITION/DESIGN STRATEGY					
6.1 Acquisition/Design Strategy:					
7. USE AND MAINTENANCE					
7.1 Software User/Administrator Authorization Process:					
7.2 Authorized Software User Requirements:					
7.3 Authorized Software Users:					
7.4 Authorized Administrator Requirements:					
7.5 Authorized Administrators:					
7.6. Risk List Required?	Yes <input type="checkbox"/>	No <input type="checkbox"/>	7.7 Risk Register Minimum Review/Frequency:		
7.8 Software Quality Assessment Required?	Yes <input type="checkbox"/>	No <input type="checkbox"/>	7.9 Minimum Assessment Frequency & Method(s):		
7.10 In-Use Testing Required?	Yes <input type="checkbox"/>	No <input type="checkbox"/>	7.11 Minimum In-Use Test Frequency & Method(s):		
7.12 Other					

## LANL

SWDS No.:

Rev.:

**8.0 ATTACHMENT LIST**

8.1 Attachment No.	8.2 Attachment Date	8.3 Attachment Title

**9.0 REVISIONS**

9.1 Revision	9.2 Date	9.3 Change Description and Reason for Change

**10. ATTACHMENTS**

--

*This search tool may not include the latest files; use individual files for work.*

## GENERAL

This software data sheet, and the documents referenced herein, provide the *software-specific* information necessary to implement the requirements of Chapter 21, *Software*. For definitions, general requirements, and processes, see SOFT-GEN.

LANL personnel: Endeavor to use Chapter forms as-is and report issues and improvement ideas to the Chapter 21 POC. POC may authorize other methods equivalent to chapter forms in writing.

LANL subcontractors must use Chapter forms to satisfy Chapter requirements for SSC software. For Non-SSC software, subs may either use their own forms or integrate, adapt, and reformat the forms; either approach is acceptable so long as key functions, data, and approvals are retained.

## HEADER

Field	Entry Information
Software Name	Enter the name of the software in the upper right hand corner of the header.
SWDS No.	The numbering scheme of this form follows that of the SRLM's document control/records management system. For software used in the operation of existing or new LANL facilities, use numbering scheme and process specified in <a href="#">AP-341-402, Engineering Document Management in Operating Facilities</a> .
Rev.	Enter the SWDS revision number. Initial issue is 0.

## 1.0 GENERAL INFORMATION

Field	Entry Information
1.1	Enter the name of the software.
1.2	Enter the technical area(s) where the software is used. If the software is used site wide, including packaging and transportation, enter "site wide".
1.3	Enter the facility number(s) where the software is used. If the software is used site wide enter "site wide".
1.4	Enter the facility (building) name(s) where the software is used. If the software is used site wide enter "site wide".
1.5	For all ES and SB-Div software (Non-SSC and SSC), enter the software identification number (SWID). <i>The SWID should be the same SWID as on the Form 2033. SWIDs are obtained in accordance with <a href="#">AP-341-402, Engineering Document Management in Operating Facilities</a>.</i> Ensure the SWID is part of the SWDS and subsequent software documentation.
1.6	For SSC software, enter the equipment identification number associated with the software per ESM Chapter 1, Section 200 (e.g., 03-0216-HVAC-BAS-1). For Non-SSC software, enter NA.
1.7	Provide (a) what the software does and (b) why it is needed (justification).
1.8	Indicate whether the software is SSC Software or non-SSC software by checking the appropriate box. Check only one box.
1.9	Indicate the highest management level (ML) associated with the software. Reference <a href="#">AP-341-502</a> . (ML-1 is the highest ML indication/has the greatest associated consequence).

## LANL

Field	Entry Information
	If the software associated with multiple MLs, check the highest ML box. Check only one box.
1.10	Indicate whether the software is safety software or non-safety software by checking the appropriate box. Reference Form <a href="#">2033</a> . If the software is used for both, check the Safety software box. Check only one box.

## 2.0 ROLES AND DATA SHEET APPROVALS

Field	Entry Information
2.1	Enter the name of the person(s) fulfilling the software role(s). For assistance in determining the SO and SRLM, see SOFT-GEN Appendix C: <i>SO and SRLM Decision Diagram for FAC-COE</i> . One person i.e., one software user (SU) etc., is a sufficient representative sample for approving the SWDS.
2.2	Enter the Z number (if available). Enter NA if not available.
2.3	Enter the date and either initials or signature that the software data sheet was reviewed and approved. Electronic signatures are acceptable. Signatures are required by the following and optional for others: SRLM, SU RLM, SU, SO, and FDAR or DA.
2.4	Enter an "x" for each role that the person is fulfilling. Note that one person may serve in multiple roles. If the person is assigned to review and approve, enter an "xs".
2.5	Enter clarifying text as necessary to clarify roles and responsibilities. See Chapter 21 for roles and responsibilities for the DA, FDAR, SD, SD RLM, SO, SRLM, SU, and SU RLM. If roles are added that are in addition to the standard roles in Chapter 21, specify the roles and responsibilities in this section. If no clarifications are necessary, enter "NA".

## 3.0 SOFTWARE PROJECT MANAGEMENT

Field	Entry Information
3.1	<p>To support software-specific project management, enter the estimated Rough Order of magnitude (ROM) software costs for the following project management/risk management activities for the next five years. Attach supporting detail as required. See SWDS example.</p> <p><b>Work Breakdown Structure (WBS) 01 - Design/Acquire Software.</b> The cost of designing and/or acquiring the software (including upgrades). Include planning, planning documentation (for hardware and software), and software quality assurance costs. Include labor, equipment and materials.</p> <p><b>WBS 02 – Acquire Hardware.</b> The cost of the associated hardware. Include computer servers, computer workstations, peripherals, etc. Include labor, equipment and materials.</p> <p><b>WBS 03 – Install/Verification and Validation (V&amp;V).</b> The cost of installation and the associated V&amp;V. Include labor and if required, equipment and materials.</p> <p><b>WBS 04 – Maintain.</b> The cost of maintaining the integrated software/hardware system. Include problem reporting, corrective action, configuration management, etc.</p> <p><b>Note:</b> Cost and schedule information are planning level estimated provided to help management allocate sufficient resources to successfully manage the software throughout its lifecycle. Detailed cost estimating is not required.</p>

### LANL



#### 4.0 REQUIREMENTS

Field	Entry Information
4.1	Enter the requirements documents in the order of precedence. (I.e., Document 1 takes precedence over document 2.) Include the document number and name and revision. For simple software applications such as that for some Off-the-Shelf (OTS) design/analysis software, the requirements may also be entered directly into this section rather than in separate requirements documents. For example, enter the required version of ASME B31.3.
4.2	Enter high-level key planning assumptions/constraints. Planning assumptions/constraints are those that may affect the overall design/acquisition approach, scope, schedule or cost.

#### 5.0 DELIVERABLES

Field	Entry Information
5.1	Enter the deliverable for the software required throughout the software life cycle. See SOFT-GEN section, Attachment B for a summary of Chapter 21 deliverables.

#### 6.0 ACQUISITION/DESIGN STRATEGY

Field	Entry Information
6.1	Describe the approach to acquiring/designing the software, support software and associated hardware. Indicate if the software is existing or new. If the software is safety software (ML-1, ML-2 or ML-3) indicate whether the software will be acquired from an NQA-1 qualified supplier or whether it will be acquired from a Non NQA-1 qualified supplier using commercial grade dedication (CGD). See acquisition factors such as cost, supplier availability in SOFT-ACQUIRE. Provide a summary level description of what work will be performed by LANL personnel and what work will be performed by supplier(s). Indicate whether the location and whether the software will be for a server-based use (generally preferred) or standalone computer use. Describe licenses, registrations and/or services contracts that must be maintained.

#### 7.0 USE AND MAINTENANCE

Field	Entry Information
7.1	Describe the person(s) and process for authorizing software users/administrators. See P781-1, <i>Conduct of Training</i> .
7.2	Enter the minimum training related requirements that a person must fulfill to be an authorized user of the software (required reading, supplier training courses, certifications, etc.). The authorized user level is as described in the software user documentation for one to use the software for its intended purpose. It does not include the authority to make changes to the computer program code or perform software maintenance activities that may be described in software maintenance manuals. See P781-1.
7.3	Enter the website location of the current (active) list of authorized users (i.e., ES-DO website), otherwise, enter the names and if available, Z#s of authorized users. If authorized

### LANL

Field	Entry Information
	users are listed in Section 2 of the SWDS, it is acceptable to enter "See SWDS Section 2".
7.4	Enter the minimum training related requirements that a person must fulfill to be an authorized administrator of the software (required reading, supplier training courses, certifications, etc.). The authorized administrator level is as described in the software user documentation for one to use the software for its intended purpose plus it includes the authority to make changes to the computer program code or perform software maintenance activities that may be described in software maintenance manuals.
7.5	Enter the website location of the current (active) list, otherwise, reference or enter the names and if available, Z#s of authorized administrators.
7.6	Indicate whether a risk list is required by checking either the yes or no box. (A risk list is required for ML-1 and ML-2 software). If required, assess and control risks based on industry accepted methods such as those described in <a href="#">DOE SQAS21.01, 06-1999, Software Risk Management a Practical Guide</a> , <a href="#">DOE G 413.3-7, Risk Management Guide</a> and/or <a href="#">DOE G 414.1-4, Safety Software Guide for Use with 10 CFR 830 Subpart A, Quality Assurance Requirements</a> , and <a href="#">DOE O 414.1C, Quality Assurance</a> . Attach the risk list.
7.7	Indicate the minimum frequency that the risk list must be reviewed and as required, revised to ensure accuracy of the list.
7.8	Indicate whether a software quality assessment is required by checking either the yes or no box.
7.9	Indicate the minimum frequency that a software quality assessment must be performed, and the planned assessment methods. Multiple assessment methods are acceptable (e.g., <a href="#">P330-3, Quality Audits per PD328, LANL Assessment Program</a> , <a href="#">AP-341-901, Performing Vital Safety System Assessments per P341, Facility Engineering Processes Manual</a> ). Software quality may be assessed as part of broader assessments.
7.10	Indicate whether in-use testing is required by checking either the yes or no box. (In-use testing is required for ML-1 and ML-2 software and recommended for ML-3 software).
7.11	Indicate the minimum frequency that in-use tests must be performed, and the acceptable test method(s). Note that in-use software tests may be an element of broader tests as long as they satisfy the software in-use test requirements.
7.12	As required, provide any other needed software-specific protocols required to promote proper software use, maintenance and retirement in this section (e.g., operational event documentation review and retention protocols, application log protocols, etc.). See SOFT-MAIN for additional information. Enter NA if not applicable or none.

### 8.0 ATTACHMENT LIST

Field	Entry Information
8.1	List attachments as appropriate. Enter the attachment number.
8.2	Enter the attachment date.
8.3	Enter the attachment title, including revision as appropriate.

## LANL



# Conduct of Engineering Software Data Sheet Form (SWDS) Instructions with Example

## 9.0 REVISIONS

Field	Entry Information
9.1	Enter the revision number here and update the revision number in the form header.
9.2	Enter the date of the revision.
9.3	Enter a summary description of what was changed and the reason for the change.

## 10.0 ATTACHMENTS

Provide attachments here.
---------------------------

*This search tool may not include the latest files; use individual files for work.*

## LANL

1.0 GENERAL INFORMATION															
1.1 Software Name:		Combustible Material Tracker (CBT) Software													
1.2 TA Number(s):	03	1.3 Facility Number(s):	2333	1.4 Facility Name(s):		Nonproliferation Security Facility (NSF)									
1.5 SWID	03-2333-045		1.6 Equipment ID:			NA									
1.7 Software Summary Description/Need:		The software is used as a tool to track combustible material. It is needed because tracking combustible material manually (without software) is less efficient and error-prone.													
1.8 SSC Software/Non-SSC Software:		SSC <input type="checkbox"/> Non-SSC <input checked="" type="checkbox"/>													
1.9 Highest ML Associated With the Software		ML-1 <input type="checkbox"/>		ML-2 <input type="checkbox"/>		ML-3 <input checked="" type="checkbox"/>		ML-4 <input type="checkbox"/>							
1.10 Software Type		Safety Software <input type="checkbox"/>			Non-Safety Software <input checked="" type="checkbox"/>										
2.0 ROLES AND DATA SHEET APPROVALS															
2.1 Name (Last, First):	2.2 Z Number:	2.3 Data Sheet Approval and Approval Date	2.4 Role:												
			ESM*	SRLM*	SD*	SD RLM*	SU*	SU RLM*	FDAR*	DA*	ISSO Rep.	Other	Other	Other	
Trujillo, Robert	124268		XS												
Lopez, Pete	121987			XS				XS							
Vigil, Mary	115970				X										
Thompson, Tom	143987					XS									
Bradley, Mike	114987						XS								
Heitkamp, Heidi	121987									XS					
Walker, Johnny	123495										X				
Not Used	NA														
2.5 Role Comments/Clarifications:		<p>1. Roles marked with a "*" are as stated in ESM Chapter 21.</p> <p>2. The ISSO Representative (Rep.) is responsible for (a) ensuring applicable PD210 and SD210 cybersecurity/security requirements are identified and reported to the SRLM and (b) supporting implementation of such requirements.</p> <p>3. Roles marked with "XS" must review, approve, date, and either initial or sign in Block 2.3. Roles marked with an "X" do not require review, approval date or signature. The X is provided to show who is filling the role.</p>													
3.0 SOFTWARE PROJECT MANAGEMENT															

**LANL**

SWDS No.: SWDS-16-03-2333-0089 Rev.: 0

3.1 Estimated Rough Order of Magnitude (ROM) Software Project Management Cost, K\$					
Fiscal Year (FY):					Total, K\$ (FY16 -20):
2016	2017	2018	2019	2020	
540	215	200	230	655	1,840
<b>4.0 REQUIREMENTS</b>					
4.1 Software Requirements and Hierarchy:		1. ESM Chapter 21, Software 2. SRS-15-03-NSF-0002 CBT Software Requirements Specification			
4.2 Key Planning Assumptions/Constraints:		1. FY16 upgrade must be completed by July 1, 2015. If not completed by that date, Los Alamos Field Office authorization is required to operate with existing software.			
<b>5. DELIVERABLES</b>					
5.1 Deliverables:		<ol style="list-style-type: none"> <li>Form 2033 and revisions.</li> <li>Documents (e.g., user manual) and computer programs as per SBL-15-03-2333-0234 (and subsequent revisions), Software Baseline for CBT with software replacement (including operating system) frequency estimated every 4 years.</li> <li>Hardware as listed in SD-15-03-NSF-0006, CBT Software Design Document; Programmable Logic Controllers and associated devices (e.g., firewalls) replacement frequency estimated every 4 years.</li> <li>Approval for use documentation (Section SOFT-V&amp;V, Att. B, as required)</li> <li>Non-SSC Software Change Packages (Section SOFT-GEN, Att. E, as required)</li> <li>Problem reporting &amp; corrective action documentation (as required)</li> <li>Institutional-Level safety software inventory and ES-DO inventory updates as required.</li> <li>Software quality assessment reports, bi-annual or as required.</li> <li>Software risk register updates, as required.</li> <li>Software data sheet (this document) revisions, as required.</li> <li>Operational event documentation</li> </ol>			
<b>6. ACQUISITION/DESIGN STRATEGY</b>					
6.1 Acquisition/Design Strategy:		<ul style="list-style-type: none"> <li><b>Primary Software:</b> The primary software is an existing C++ code. This code will undergo re-design by qualified LANL programmers and include software code/deliverables as required by ESM Chapter 21 for ML-3 software.</li> <li><b>Support Software:</b> Support software is off-the-shelf software (e.g., MS 2003 and MS Windows XP) that will be acquired as ML-4 software from <a href="#">LANL Electronic Software Distribution (ESD)</a>.</li> </ul>			

## LANL

SWDS No.: SWDS-16-03-2333-0089 Rev.: 0

		<ul style="list-style-type: none"> <li><b>Hardware:</b> Hardware is conventional PLC related hardware and peripherals that will be procured as ML-4 hardware.</li> </ul>			
<b>7. USE AND MAINTENANCE</b>					
7.1 Software User/Administrator Authorization Process:		<ul style="list-style-type: none"> <li>The SRLM will identify, evaluate and authorize software users and administrators based on need and the knowledge, skills and abilities required as indicated in this document. Authorized personnel are indicated in this document.</li> </ul>			
7.2 Authorized Software User Requirements:		<ul style="list-style-type: none"> <li>Online Utrain Course No. 45458, UM-15-03-NSF-001, <i>CBT User Manual</i>, and</li> <li>Online Utrain Course No. 52346, AP-15-03-NSF-0007, <i>CBT Administrative Procedure</i>.</li> </ul>			
7.3 Authorized Software Users:		<ul style="list-style-type: none"> <li>As listed on ES-DO website.</li> </ul>			
7.4 Authorized Administrator Requirements:		<ul style="list-style-type: none"> <li>Authorized Software User Requirements, plus</li> <li>Online Utrain Course No. 42341, MM-15-03-NSF-0009, <i>CBT Maintenance Manual</i>, and</li> <li>Minimum of five (5) years of experience in C++ programming</li> </ul>			
7.5 Authorized Administrators:		<ul style="list-style-type: none"> <li>SO, SD (see Section 2 for names and Z#s)</li> </ul>			
7.6. Risk List Required?		Yes <input type="checkbox"/>	No <input checked="" type="checkbox"/>	7.7 Risk List Minimum Review/Revision Frequency:	Not Applicable
7.8 Software Quality Assessment Required?		Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	7.9 Minimum Assessment Frequency & Method(s):	Bi-annual; <a href="#">P330-3</a> , <i>Quality Audits</i>
7.10 In-Use Testing Required?		Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>	7.11 Minimum In-Use Test Frequency & Method(s):	Annual; SWTP-15-03-NSF-0008, <i>CBT Software Test Plan</i>
7.12 Other		<ol style="list-style-type: none"> <li>The CBT software produces various operational event documentation, including self-diagnostic test results, as described in the O&amp;M documents. As part of the planned assessments review the documentation for issues or issue precursors and take actions as required. The operational event documentation file should be retained for a minimum of four years within the computer program and backed up to the DCS02 server at least annually. The documentation is not a quality record.</li> </ol>			
<b>8.0 ATTACHMENT LIST</b>					
8.1 Attachment No.	8.2 Attachment Date	8.3 Attachment Title			
A1	06/17/2016	CBT Software Cost Estimate Supporting Detail, Rev. 0			
<b>9.0 REVISIONS</b>					

## LANL



**Conduct of Engineering  
Software Data Sheet (SWDS)  
Example for CBT Software**

SWDS No.: SWDS-16-03-2333-0089      Rev.: 0

9.1 Revision	9.2 Date	9.3 Change Description and Reason for Change
0	09/17/2016	Original Issue

**10.0 ATTACHMENTS**

**Attachment A1: 06/17/2016, CBT Cost Estimate Supporting Detail, Rev. 0**

Item No:	Item Name:	Estimated Rough Order Of magnitude (ROM) Cost, K\$:					Total, K\$ (FY16 – FY20):
		Fiscal Year (FY):					
		2016	2017	2018	2019	2020	
1	Design/Acquire Software	200	0	0	0	235	435
2	Acquire Hardware	0	0	0	0	25	25
3	Install/V&V	150	0	0	0	175	325
4	Maintain	190	215	200	230	220	1,055
<b>Total</b>		<b>540</b>	<b>215</b>	<b>200</b>	<b>230</b>	<b>655</b>	<b>1,840</b>

**Key Cost & Schedule Reporting Clarifications:**

1. Costs are fully burdened at 25% of direct costs and escalated at 4% per year.
2. Design/Acquire Software costs include documentation update costs.
3. Software upgrade costs includes cost to upgrade existing version of Combustible CBT server and workstation support software (e.g., Windows Server 2016 and MS Office Outlook) to newer versions.
4. Maintenance cost includes 0.5 FTE SO Level of Effort (LOE) at \$180 /hour (burdened); and a bi-annual software quality assessment at an average \$20k per assessment. Maintenance costs do not include annual supplier service contract maintenance, notifications knowledge base updates.
5. Hardware costs include costs for servers, workstations, field scanners, firewall, and router see hardware listing in SWSD-15-03-NSF-0006.
6. All work is planned to begin in 1<sup>st</sup> quarter of shown FY and end in third-quarter of same FY to allow for a three month schedule contingency within each FY.
7. The SO will report issues of concern/deviations in scope, schedule or cost that adversely affect the software plan to the SRLM as part of the reporting processes associated with use of the software (e.g., issues with software scope, schedule or cost for software used in the facility design/analysis of facilities is reported as part of the project/modification that requires the design/analysis).
8. FY16 hardware costs covered by ES Division Office as part of ES Server 0294 software FY16 upgrade.

**LANL**

TABLE OF CONTENTS

1.0 PURPOSE, SCOPE, AND APPLICABILITY ..... 1

2.0 KEY DEFINITIONS AND ACRONYMS ..... 4

3.0 SSC SOFTWARE ACQUISITION ..... 4

3.1 ML-4 SSC Software ..... 4

3.2 ML-1 through ML-3 SSC Software ..... 4

4.0 NON-SSC SOFTWARE ACQUISITION ..... 7

4.1. Existing Non-SSC Software (All MLs) ..... 7

4.2. New ML-4 Non-SSC Software ..... 8

4.3. New Software used in SSC Design that is Simple and Easily Understood ..... 8

4.4. Other New Non-SSC Software ..... 8

Rev	Date	Description	POC	RM
0	06/23/16	Initial issue	Tobin Orin, ES-DO	Mel Burnett, CENG-OFF

1.0 PURPOSE, SCOPE, AND APPLICABILITY

This section, SOFT-ACQUIRE, describes the “how, when, and who” for acquiring software and software services. See Table 21.2-1 for a summary of this section.

See Chapter 21 applicability in SOFT-GEN in addition:

1. This section applies to workers that are acquiring software.<sup>1</sup>
2. To apply this section, the software must first be identified and a draft or final software determination (Form 2033) completed. See SOFT-GEN.
3. **Freeware:** Freeware cannot be acquired without the approval of the Chapter POC or designate(s).<sup>2</sup> Exceptions: Uncomplicated software tools (see Chapter 21 applicability in SOFT-GEN), toolbox codes, “Simple and Easily Understood” software used in the design of SSCs, software from other organizations within the DOE complex, and/or software acquired from other government agencies (e.g., US EPA, Army Corps of Engineers). Note: P1011, *Computer Hardware, Software, and Services: Exceptions to Acquisition, Maintenance, and Support* has additional requirements for freeware.
4. See SOFT-V&V for verification and validation (V&V), including commercial grade dedication; see SOFT-MAINT for maintenance of acquired software.
5. Software acquisition includes associated software services throughout this section.

<sup>1</sup> Thus, software developed by a LANL owner (self-developed) would not follow this Section, but would follow all others.

<sup>2</sup> Should be necessary for LANL mission accomplishment and where no feasible alternative solutions are available.



2 Section SOFT-ACQUIRE: Software Acquisition

Rev. 0, 06/23/16

Table 21.2-1 SOFT-ACQUIRE Section Summary for SSC and Non-SSC Software (This is a summary only and does not include all requirement details. See text for details.)										
Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			Reference	
		1	2	3	4	How	When	Who <sup>2,3</sup>	P1040 R9 Ref. Section	ESM Ch. 21 Ref. <sup>8</sup>
<b>I. For SSC Software</b>										
1	Acquire new software	R	R	G	G	<ul style="list-style-type: none"> <li>Evaluate and document whether to acquire from NQA-1 -- or Non-NQA-1 supplier and commercially dedicate (CGD)<sup>4</sup></li> <li>Develop a SWDS, (SOFT-GEN-FM03)<sup>5</sup></li> <li>Develop Statement of Work (SOW)<sup>6</sup></li> <li>Acquire per <a href="#">P840-1</a>, <a href="#">PD-210</a> and <a href="#">SD-210</a></li> <li>Develop P-card or Purchase Order (PO)</li> <li>Develop interface agreement, integrated contract orders, interagency agreements or equivalent<sup>7</sup></li> <li>Complete <a href="#">Exhibit H</a><sup>5</sup></li> </ul>	<ul style="list-style-type: none"> <li>When needed to perform work</li> <li>Prior to use</li> </ul>	<ul style="list-style-type: none"> <li>SO (D)</li> <li>SRLM (R, A)</li> </ul>	3.3.4.a	SOFT-ACQUIRE, 3
<b>II. For Non-SSC Software</b>										
1	Acquire existing LANL software	R	R	R	G	<ul style="list-style-type: none"> <li>Look for, evaluate and use existing software if suitable</li> <li>Acquire in accordance with the SRLM instructions and as applicable, SWDS</li> <li>Update authorized users/SWDS as required</li> </ul>	<ul style="list-style-type: none"> <li>When needed to perform work</li> <li>Prior to use</li> </ul>	<ul style="list-style-type: none"> <li>SU (D)</li> </ul>	3.3.4.a	SOFT-ACQUIRE, 4
2	Acquire new software	R	R	G	G	<ul style="list-style-type: none"> <li>Determine and document if software is used in SSC design and is simple and easily understood; document and verify with each use</li> <li>Determine if software is a toolbox code</li> <li>Evaluate and document whether to acquire from NQA-1; or Non-NQA-1 supplier and commercially dedicate the software (CGD)<sup>4</sup></li> <li>Develop SWDS, (SOFT-GEN-FM03)<sup>5</sup></li> <li>Develop Statement of Work (SOW)<sup>6</sup></li> <li>Acquire per <a href="#">P840-1</a>, <a href="#">PD-210</a> and <a href="#">SD-210</a></li> <li>Develop CGD documentation per <a href="#">AP-341-703</a></li> <li>Develop P-card or Purchase Order (PO)</li> <li>Develop inter- or intra- LANL agreements<sup>7</sup></li> <li>Complete <a href="#">Exhibit H</a><sup>5</sup></li> </ul>	<ul style="list-style-type: none"> <li>When needed to perform work</li> <li>Prior to use</li> </ul>	<ul style="list-style-type: none"> <li>SO (D)</li> <li>SRLM (R, A)</li> </ul>	3.3.4.a	SOFT-ACQUIRE, 4

This search tool may not include the latest files; use individual files for work.

2 Section SOFT-ACQUIRE: Software Acquisition

Rev. 0, 06/23/16

**Table 21.2-1 SOFT-ACQUIRE Section Summary for SSC and Non-SSC Software**  
 (This is a summary only and does not include all requirement details. See text for details.)

Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			Reference	
		1	2	3	4	How	When	Who <sup>2,3</sup>	P1040 R9 Ref. Section	ESM Ch. 21 Ref. <sup>8</sup>
<p><sup>1</sup> ML = Associated Management Level per Form <a href="#">2033</a>. Complete draft Form <a href="#">2033</a> prior to acquisition. R = Required. G = Required but graded.</p> <p><sup>2</sup> D = Develop or implement; R = Review; A = Approve.</p> <p><sup>3</sup> SO = Software Owner. SRLM = Software Responsible Line Manager. SU = Software User</p> <p><sup>4</sup> CGD applies to safety software only.</p> <p><sup>5</sup> Required for ML-1 through ML-3 (safety and risk significant) software or for any ML if the supplier is required to come onsite to install/maintain the software.</p> <p><sup>6</sup> Applies to ML-1 through ML-3 software only.</p> <p><sup>7</sup> See <a href="#">P840-1</a> for details. Required for acquisitions from LANL organizations, DOE sites, or government agencies.</p> <p><sup>8</sup> Ref. = ESM Chapter 21 section name and subsection number (e.g., Section SOFT-GEN, Subsection 3, Software Identification and Determination).</p>										

This search tool may not include the latest files; use individual files for work.

## 2 Section SOFT-ACQUIRE: Software Acquisition

Rev. 0, 06/23/16

## 2.0 KEY DEFINITIONS AND ACRONYMS

See SOFT-GEN for key definitions and acronyms.

## 3.0 SSC SOFTWARE ACQUISITION

## 3.1 ML-4 SSC Software

A. How.

1. Acquire in accordance with [P840-1](#), *Quality Assurance for Procurements*, [PD210](#), *Cyber Security Program*, [SD210](#), *Information Risk Management Framework* and the following:
  - a. Procure with a purchase card (P-card) or purchase order (PO) using LANL online systems. Specify software requirements and acceptance criteria.
  - b. Complete an [Exhibit H](#) if the supplier comes onsite to install and/or maintain the software (ref. [P840-1](#)) as required.
  - c. *Guidance: For software acquired from other organizations within LANL or within the DOE complex, use interface agreements, integrated contracts or equivalent where possible. See [P840-1](#) for details and an example interface agreement.*
  - d. Protect against suspect/counterfeit items (S/CI) items by procuring from reputable authorized suppliers and ensuring valid software licenses. See [P840-9](#), *Suspect Counterfeit Items (S/CI)* for additional information.
  - e. Retain acquisition documentation in accordance with the SRLM's governing records management process.

**Note:** Ensure the correct SWID (obtained when completing Form 2033) is part of the acquisition documentation record number. See [AP-341-402](#), *Engineering Document Management in Operating Facilities* for details.

B. When. When software is needed to perform work.C. Who. The SO develops and the SRLM reviews and approves the acquisition<sup>3</sup>.

## 3.2 ML-1 through ML-3 SSC Software

Software for ML-1 through ML-3 applications must be acquired in accordance with this Subsection.

<sup>3</sup> Additional review and approval by the requestor's Division Leader and OICO may be required by Acquisition Services Management (ASM). See [ASM 3400.00.0410](#), *Goods or Services Requiring Special Review/Approval* and/or contact an ASM representative.

## 2 Section SOFT-ACQUIRE: Software Acquisition

Rev. 0, 06/23/16

A. How.

1. Safety Software: Acquire from an NQA-1 qualified supplier, or, if from a Non-NQA-1 qualified supplier, commercial-grade-dedicate the software.
  - a. Determine whether to acquire from an NQA-1 qualified supplier or a Non-NQA-1 qualified supplier by incorporating the following factors:
    - i. the availability of NQA-1 qualified suppliers (see the [LANL Institutional Evaluated Supplier List, \(IESL\)](#));
    - ii. performance history;
    - iii. cost, schedule and other commercial factors applicable to the software lifecycle.
2. Develop a software data sheet (SWDS). Use SOFT-GEN-FM03, *Software Data Sheet Form (SWDS)*. Document the NQA-1/Non-NQA-1 acquisition strategy on the SWDS. As required by the SRLM, document supporting rationale on the SWDS or elsewhere.
3. Risk Significant software: Acquire from an NQA-1 qualified supplier or a Non-NQA-1 qualified supplier. CGD does not apply.
4. Develop a Statement of Work (SOW). Use the applicable SOW forms (from ES-EPD, AP-341-702, or equivalent). *Guidance: [ASM website](#)*. Address the following requirements:

**Note:** Treat bold text (not headings) as ML-1 and ML-2 requirements but guidance for ML-3 throughout SOFT-ACQUIRE. Plain text is a requirement for ML-1 through ML-3.

- a. Identify the software requirements, **including but not limited to specific functionality in specified environments, acceptance criteria, and supplier/LANS interfaces**; for designed software, see SOFT-DESIGN.
- b. Require documentation (and/or computer programs) describing methods used in developing and validating the software (e.g., test cases). See SOFT-DESIGN and SOFT-V&V.
- c. **From NQA-1 qualified suppliers, and when available from Non-NQA-1 qualified suppliers, require the supplier to report software errors to the purchaser, and as appropriate, the ability for the purchaser to report software errors to the supplier.**
  - d. Specify software features, including requirements for safety, security functions and performance as applicable.
- e. Specify requirements for supplier notification of new releases or other issues (in addition to errors) that impact the software operation.
- f. **As required, specify mechanisms for the users to obtain assistance in operating the software.**
- g. Specify quality assurance requirements (e.g., conform to DOE Order 414.1D or ASME NQA-1-2008/NQA-1A-2009 or [ASME NQA-1-2008/NQA-1A-2009](#), *Quality Assurance Requirements for Nuclear Facility Applications, Part I and Part II*). Specify quality assurance requirements on an Exhibit H for ML-1 through ML-3 (safety and risk

## 2 Section SOFT-ACQUIRE: Software Acquisition

Rev. 0, 06/23/16

significant) software, and for any ML software if the supplier is required to be onsite to install/maintain the software.

h. Address the following guidance:

- 1) For safety software that will undergo CGD, specify required supplier support to CGD the software (e.g., access for supplier assessments, surveys, 3<sup>rd</sup> party dedicators).
- 2) Specify training and/or technical support required to promote successful installation and use of the software.
- 3) Consider specifying minimum acceptable performance history for similar applications.
- 4) Specify desired method/media for computer program delivery (e.g., electronic download, compact disc).
- 5) Consider attaching marked-up vendor product information that identifies key software requirements (e.g., 64 bit, Windows 10 operating system, Delta V version 10.2).
- 6) As needed, contact the Office of Chief Information Officer (OCIO) for assistance implementing [PD210](#) and/or [SD210](#).

5. As required, clarify LANL-supplier responsibility on the SWDS.

Example: LANL was planning to acquire software for an ML-2 application. The supplier provided notifications of defects, new releases or other issues that may impact the software operation on the supplier's website. The SWDS specified that the SRLM would periodically review the supplier website and/or contact the supplier to ensure the notification requirement was satisfied.

6. Acquire in accordance with [P840-1](#), [PD210](#), and [SD210](#). Note the following:

- a. For intra-laboratory acquisitions use interface agreements per [P840-1](#). See [P840-1](#) for an example interface agreement. Address SOW requirements in the agreement.
- b. For inter-laboratory acquisitions from other organizations within the DOE complex, use integrated contract orders (ICOs) per [P840-1](#). Address SOW requirements in the ICO.
- c. P-cards cannot be used for purchasing software associated with ML-1 through ML-3 applications (includes safety software and risk significant software).
- d. Protect against suspect/counterfeit items (S/CI) items by procuring from reputable suppliers and ensuring valid software licenses. See [P330-9](#), *Suspect Counterfeit Items (S/CI)* for additional information.
- e. Retain acquisitions documentation in accordance with the SRLM's governing records management process.  
  
Ensure the correct SWID (obtained when completing Form 2033) is part of the acquisition documentation record number. See [AP-341-402](#) for details.
- f. See LANL Institutional Evaluated Supplier List ([IESL](#)) for a listing of NQA-1 qualified suppliers.

## 2 Section SOFT-ACQUIRE: Software Acquisition

Rev. 0, 06/23/16

- g. See [AP-341-703](#), *Commercial Grade Dedication* for CGD. (See SOFT-V&V for use of performance history for dedicating computer programs.) Identify and control acquired software during the dedication process.
- B. When. Acquire when software is needed to perform work. Complete acquisition prior to approval for use as described in SOFT-V&V.
- C. Who. The SO performs acquisition activities. The SRLM must review and approve software acquisitions.<sup>4</sup>

## 4.0 NON-SSC SOFTWARE ACQUISITION

Look for, evaluate, and if suitable, use existing software before acquiring new software. If existing software cannot be used, then determine if the software is used in SSC design that is simple and easily understood; or other software. Then, acquire in accordance with the following applicable text.

## 4.1. Existing Non-SSC Software (All MLs)

- A. How.
  - 1. Look for existing software at LANL that may be suitable. *Review the Engineering Services Division software inventory. Contact the ES-Div Software Coordinator and/or subject matter expert (SME) that may be using the desired software or similar software. Coordinator should be listed on ES-Div and Ch 21 webpages.*
  - 2. Evaluate whether existing software is suitable for the intended application. Evaluate limitations (including licenses), boundary conditions, performance (including issues/risks), and bases. Incorporate the following in the evaluation:
    - a. Ensure the ML of the existing software is commensurate with the intended use. *For example, if CAESAR II software were acquired and managed as ML-3 software, it can only be used on ML-3 and ML-4 applications. It cannot be used on ML-1 or ML-2 applications unless: (a) it were determined to be simple and easily understood software used in design, or (2) it were commercially grade dedicated.*
    - b. Ensure the industry code associated with the software is appropriate for the intended application. *For example, CAESAR II, version 7.0 is based on the 2012 edition of American Society of Mechanical Engineers (ASME) B31.3; the software should be used only for designs where this edition of the code is appropriate.*
    - c. For ML-1 through ML-3 software, update authorized users, SWDS as required.
- B. When. Evaluate and use existing software, if suitable, before acquiring new software. Acquire when needed to perform work.
- Who. The software user (SU) completes required acquisition actions.

---

<sup>4</sup> See previous footnote regarding other approvals such as ASM.

## 2 Section SOFT-ACQUIRE: Software Acquisition

Rev. 0, 06/23/16

## 4.2. New ML-4 Non-SSC Software

See Subsection 3.1 and apply to Non-SSC ML-4 software.

## 4.3. New Software used in SSC Design that is Simple and Easily Understood

A. How.

1. Determine if the software is in the category of software that is used in SSC design that is simple and easily understood. Such software is excluded from commercial grade dedication (CGD) or the requirement to acquire from an NQA-1 supplier. The Responsible Engineer (RE) completes the determination using the following criteria. Each criterion must be satisfied.
  - a. The software is used in the design of SSCs;
  - b. The results of the computer program can be easily confirmed through hand calculations;
  - c. A person technically qualified, as determined by the SRLM, in the subject can review and understand the program and the supporting calculations; and,
  - d. The software can be individually verified with each use (e.g., calculation). See SOFT-V&V for individual verification requirements.
2. The RE documents the determination in the calculation where the software was used and the determination is reviewed and approved as part of the calculation in accordance with SOFT-V&V.
3. If the software is in this category, then acquire the software as ML-4 software in accordance with Subsection 4.2.
4. If the software is not in this category, then acquire in accordance with Subsection 4.4.

**Note:** If the software is safety software, frequent use of the software may justify CGD or acquisition from an NQA-1 supplier.

B. When. Acquire when software is needed to perform work.

C. Who. The SO develops and the SRLM reviews and approves the acquisition (see footnote for Paragraph 3.1.c regarding ASM).

## 4.4. Other New Non-SSC Software

Software for ML-1, ML-2, and/or ML-3 applications that is not “Simple and Easily Understood” software used for SSC design must be acquired in accordance with this section. This includes toolbox codes. All of the text in this section applies to ML-1 and ML-2 software.

A. How.

Using the requirements of this section determine whether to acquire software from:

- the [DOE Safety Software Quality Assurance Central Registry](#) (hereafter Registry, *currently safety basis software only*);
- an NQA-1 qualified supplier, or
- a Non-NQA-1 qualified supplier.

## 2 Section SOFT-ACQUIRE: Software Acquisition

Rev. 0, 06/23/16

2. Toolbox codes: Determine if the software is listed on the Registry. If listed, it is a toolbox code. *Guidance:*<sup>5</sup> *Acquire in accordance with the following:*
  - a. *If the software is from other DOE site, acquire using an integrated contract order (ICO) in accordance with [P840-1](#) and this section.*
  - b. *If the software is from another government agency, acquire using an inter-agency agreement (IAA) in accordance with [P840-1](#) and this section.*
  - c. *Only the version of the toolbox code listed on the Registry can be acquired without performing CGD or acquiring from an NQA-1 supplier.*
3. Excluding toolbox codes, acquire safety software from an NQA-1 qualified supplier -- or from a Non-NQA-1 qualified supplier and commercial grade dedicate the software.
  - a. Determine whether to acquire from an NQA-1 qualified supplier or a Non-NQA-1 qualified supplier by incorporating the following factors:
    - i. the availability of NQA-1 qualified suppliers (see the [LANL Institutional Evaluated Supplier List, \(IESL\)](#));
    - ii. performance history;
    - iii. cost, schedule and other commercial factors applicable to the software lifecycle.
4. Develop a SWDS. Use SOFT-GEN-FM03, *Software Data Sheet Form (SWDS)*. Document the NQA-1/Non-NQA-1 acquisition strategy on the SWDS. As required by the SRM, document supporting rationale on the SWDS or elsewhere.
5. Risk Significant Software: Acquire from an NQA-1 qualified supplier or a Non-NQA-1 qualified supplier. CGD does not apply.
6. Develop a Statement of Work (SOW). Use the applicable SOW forms (from ES-EPD, AP-349-702, or equivalent). *Guidance:* [ASM website](#). Address the following requirements:

**Note:** Treat bold text (not headings) as ML-1 and ML-2 requirements but guidance for ML-3 throughout SOFT-ACQUIRE. Plain text is a requirement for ML-1 through ML-3.

- a. Identify the software requirements. Include as specific functionality in specified environments (as applicable), acceptance criteria, **and supplier/LANL interfaces.**
- b. Require documentation (and/or computer programs) describing methods used in **developing and** validating the software (e.g., test cases).
- c. **From NQA-1 qualified suppliers, and when available from Non-NQA-1 qualified suppliers, require the supplier to report software errors to the purchaser, and as appropriate, the ability for the purchaser to report software errors to the supplier.**

<sup>5</sup> P840-1 r8 doesn't address the central registry and P1040 r9 excludes toolbox codes from acquisition requirements.



## 2 Section SOFT-ACQUIRE: Software Acquisition

Rev. 0, 06/23/16

- d. Specify software features, including requirements for safety, security functions and performance as applicable.
- e. Specify requirements for supplier notification of new releases or other issues (in addition to errors) that impact the software operation.
- f. *As required, specify mechanisms for the users to obtain assistance in operating the software.*
- g. Specify quality assurance requirements (e.g., conform to DOE Order 414.1D or ASME NQA-1-2008/NQA-1A-2009, *Quality Assurance Requirements for Nuclear Facility Applications, Part I and Part II*). Specify quality assurance requirements on an Exhibit A for ML-1 through ML-3 (safety and risk significant) software and for any ML software if the supplier is required to be onsite to install/maintain the software.
- h. For software that includes models (see SOFT-GEN for definition and SOFT-DESIGN for additional information), ensure the appropriateness of the model for the application in addition to the appropriateness of the computer code used to provide the outcome of the model.
- i. *Address the following guidance:*
- 1) *Specify minimum design codes for the software. As a fictitious example, SAP2000 must satisfy ASCE 10-97 for steel frame design and ACI 318-08 for concrete frame design.*
  - 2) *For safety software that will undergo CGD, specify supplier support that may be required to dedicate the software (e.g., access for supplier assessments, surveys, third-party dedicators).*
  - 3) *Specify training and/or technical support required to promote successful installation and use of the software.*
  - 4) *Specify whether single user, multiuser server-based, or cloud-based licenses are required (contact the ES-Div Software Coordinator for support).*
  - 5) *Consider specify minimum acceptable performance history for similar applications.*
  - 6) *Specify desired method/media of computer program delivery (e.g., electronic download, compact disc).*
  - 7) *Consider attaching marked-up vendor product information that identifies the key software requirements (e.g., 64 bit, Windows 10, ASME B31.3 2012 Edition).*
  - 8) *As needed, contact the Office of Chief Information Officer (OCIO) for assistance implementing [PD210](#) and/or [SD210](#).*
7. As required, clarify LANL-supplier division of responsibility on the SWDS.
- Example: LANL was planning to acquire software in an ML-2 application. The supplier provided notifications of defects, new releases or other issues that may impact the software operation on the supplier's website, rather than a*

## 2 Section SOFT-ACQUIRE: Software Acquisition

Rev. 0, 06/23/16

more direct notification such as email. The SWDS specified that the SRLM will periodically review the supplier website and/or contact the supplier for the information and thereby ensure the requirement was satisfied.

8. Acquire in accordance with [P840-1](#), [PD210](#), and [SD210](#). Note the following:  
See LANL Institutional Evaluated Supplier List ([IESL](#)) for a listing of NQA-1 qualified suppliers.
  - a. See [AP-341-703](#), *Commercial Grade Dedication* for CGD. (See SOFT-V&V for use of performance history for dedicating computer programs.)
  - b. *Guidance: Intra-laboratory acquisitions should use interface agreements per [P840-1](#). See [P840-1](#) for an example interface agreement. Address SOW requirements in the agreement.*
  - c. For inter-laboratory acquisitions from other organizations within the DOE complex, use integrated contract orders (ICOs) per [P840-1](#). Address SOW requirements in the ICO.
  - d. Note that P-cards cannot be used for purchasing software associated with ML-1 through ML-3 applications (includes safety software and risk significant software).
  - e. Protect against suspect/counterfeit items (S/CI) items by procuring from reputable authorized suppliers and ensuring valid software licenses. See [P330-9](#), *Suspect Counterfeit Items (S/CI)* for additional information.
  - f. Retain acquisitions documentation in accordance with the SRLM's governing records management process.

Ensure the correct SWID (obtained when completing Form 2033) is part of the acquisition documentation record number. See [AP-341-402](#) for details.

- B. When. Acquire when software is needed to perform work. Complete acquisition prior to approval for use as described in SOFT-V&V.
- C. Who. The SO performs acquisition activities. The SRLM must review and approve software acquisitions (See previous footnote regarding other approvals such as ASM.).

**TABLE OF CONTENTS**

1.0 PURPOSE, SCOPE, AND APPLICABILITY ..... 1

2.0 KEY DEFINITIONS AND ACRONYMS.....5

3.0 SSC SOFTWARE DESIGN REQUIREMENTS (INPUT).....5

4.0 SSC SOFTWARE DESIGN (OUTPUT) .....7

    4.1 General Software Design Requirements.....7

    4.2 Software System Architectural Design.....9

    4.3 Software Requirements Specification (SWRS) .....9

    4.4 Software Architectural Design .....9

    4.5 Software Detail Design .....10

    4.6 Software System Hazard Analysis and Mitigation.....10

    4.7 Software Design Traceability and Testability .....11

    4.8 Operations and Maintenance (O&M) Instructions .....11

    4.9 SOFT-GEN Requirements and Deliverables.....11

5.0 SSC SOFTWARE COMPUTER PROGRAM LANGUAGE (CODE) .....12

6.0 NON-SSC SOFTWARE DESIGN REQUIREMENTS (INPUT) .....12

7.0 NON-SSC SOFTWARE DESIGN (OUTPUT) .....14

8.0 NON-SSC SOFTWARE PROGRAM LANGUAGE (CODE) .....14

REFERENCE: SOFTWARE DESIGN DOCUMENTATION EXAMPLE

Rev	Date	Description	POC	RM
0	06/23/16	Initial issue	Tobin Oruch, ES-DO	Mel Burnett, CENG-OFF

**1.0 PURPOSE, SCOPE, AND APPLICABILITY**

This section describes the “how, when, and who” for designing (developing) software. See Tables 21.3-1 and 21.3-2 for a summary of this section for SSC software and Non-SSC software respectively.

See Chapter 21 applicability in SOFT-GEN. In addition:

- This section does not apply to “simple and easily understood” software used in the design of SSCs if individually verified as described in SOFT-V&V.
- See SOFT-V&V for required verification and validation activities and deliverables for designed software.
- See SOFT-GEN for additional requirements that apply during design.

3 Section SOFT-DESIGN: Software Design

Rev. 0, 06/23/16

Table 21.3-1 SOFT-DESIGN Section Summary for SSC Software (This table is a summary only and does not include all requirement details. See text for details.)										
Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			Reference	
		1	2	3	4	How	When	Who <sup>2,3</sup>	P1040 R9 Ref. Section	ESM Ch. 21 Ref. Section <sup>4</sup>
1	Develop software design requirements (input)	R	R	G	G	<ul style="list-style-type: none"> <li>▪ Develop software design input deliverables:                             <ul style="list-style-type: none"> <li>○ SSC drawings (e.g., PFDs, P&amp;IDs, SOOs, hardware/instrument location drawing, location drawing, instrument/device list).</li> <li>○ SSC performance specification</li> <li>○ SSC FDD or SDD</li> <li>○ Other SSC design inputs/outputs as required</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▪ Earliest practical time</li> <li>▪ 60%, 90%, 100% SSC detail design</li> <li>▪ Prior to software design as much as practical</li> </ul>	<ul style="list-style-type: none"> <li>▪ SSC DA (D, R, A)</li> <li>▪ LANL SRLM (R, A)</li> <li>▪ FDAR (R, A)</li> </ul>	3.3.4.b	SOFT-DESIGN, 3
2	Design the software (output)	R	R	G	G	<ul style="list-style-type: none"> <li>▪ Design the software (output) deliverables:                             <ul style="list-style-type: none"> <li>○ software specification (SIVRS)</li> <li>○ software design (SWDD)</li> <li>○ software system hazard analysis and mitigation documentation (SWHA)</li> <li>○ software design traceability documentation (SWTM)</li> <li>○ operations and maintenance (O&amp;M) instructions</li> <li>○ SOFT-GEN deliverables</li> <li>○ Other software design deliverables as required</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▪ Earliest practical time</li> <li>▪ If in scope of SSC design, software requirements (draft) at 30% design submittals, firmed up at 60%.</li> <li>▪ If not in scope of SSC design, then in detail design submittals</li> <li>▪ Prior to software coding as much as practical</li> </ul>	<ul style="list-style-type: none"> <li>▪ SD (D, R, A)</li> <li>▪ SRLM (R, A)</li> <li>▪ LANL SRLM (R, A)</li> </ul>	3.3.4.b	SOFT-DESIGN, 4
3	Translate the software design into computer program language	R	R	G	G	<ul style="list-style-type: none"> <li>▪ Develop computer program language (code):                             <ul style="list-style-type: none"> <li>○ data structures/files,</li> <li>○ source code (where feasible),</li> <li>○ executable code</li> </ul> </li> <li>▪ Provide computer program listings</li> </ul>	<ul style="list-style-type: none"> <li>▪ Earliest practical time</li> <li>▪ Prior to final acceptance testing</li> </ul>	<ul style="list-style-type: none"> <li>▪ SD (D, R, A)</li> <li>▪ SRLM (R, A)</li> <li>▪ LANL SRLM (R, A)</li> </ul>	3.3.4	SOFT-DESIGN, 5

This search tool may not include the latest files; use individual files for work.

3 Section SOFT-DESIGN: Software Design

Rev. 0, 06/23/16

Table 21.3-1 SOFT-DESIGN Section Summary for SSC Software (This table is a summary only and does not include all requirement details. See text for details.)										
Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			Reference	
		1	2	3	4	How	When	Who <sup>2,3</sup>	P1040 R9 Ref. Section	ESM Ch. 21 Ref. Section <sup>4</sup>
<p><sup>1</sup> ML = Associated Management Level per Form <a href="#">2033</a>. R = Required. G = Required but graded.</p> <p><sup>2</sup> D = Develop or implement; R = Review; A = Approve.</p> <p><sup>3</sup> SSC DA = Design agency for SSC that is associated with the software); SRLM = Software Responsible Line Manager (prior to turnover to LANL). LANL SRLM = SRLM after turnover to LANL. FDAR = Facility Design Authority Representative.</p> <p><sup>4</sup> Ref. = <a href="#">ESM</a> Chapter 21 section name and subsection number (e.g., section SOFT-GEN, subsection 3, Software Identification and Determination).</p>										

Table 21.3-2 SOFT-DESIGN Section Summary for Non-SSC Software (This table is a summary only and does not include all requirement details. See text for details.)										
Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			Reference	
		1	2	3	4	How	When	Who <sup>2,3</sup>	P1040 R9 Ref. Section	<a href="#">ESM</a> Ch. 21 Ref. <sup>5</sup>
1	Develop software design requirements (input)	R	R	G	G	<ul style="list-style-type: none"> <li>▪ Develop system requirements (input) specification (SWYRS)</li> </ul>	<ul style="list-style-type: none"> <li>▪ Earliest practical time</li> <li>▪ Prior to software design as much as practical</li> </ul>	<ul style="list-style-type: none"> <li>▪ SO (D, R, A)</li> <li>▪ SRLM (R, A)</li> <li>▪ FDAR (R, A)</li> </ul>	3.3.4.b	SOFT-DESIGN, 6
2	Design the software (output)	R	R	G	G	<ul style="list-style-type: none"> <li>▪ Design the software (output) deliverables:                             <ul style="list-style-type: none"> <li>○ software specification (SWRS)</li> <li>○ software design (SWDD)</li> <li>○ software system hazard analysis and mitigation documentation (SWHA)</li> <li>○ software design traceability documentation (SWTM)</li> <li>○ SOFT-GEN deliverables</li> <li>○ Other software design deliverables as required</li> <li>○ Computer model mathematical terms<sup>4</sup></li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>▪ Earliest practical time</li> <li>▪ Prior to software coding as much as practical</li> </ul>	<ul style="list-style-type: none"> <li>▪ SD (D, R, A)</li> <li>▪ SRLM (R, A)</li> </ul>	3.3.4.b	SOFT-DESIGN, 7

This search tool may not include the latest files. Use individual files for work.

3 Section SOFT-DESIGN: Software Design

Rev. 0, 06/23/16

Table 21.3-2 SOFT-DESIGN Section Summary for Non-SSC Software (This table is a summary only and does not include all requirement details. See text for details.)										
Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			Reference	
		1	2	3	4	How	When	Who <sup>2, 3</sup>	P1040 R9 Ref. Section	ESM Ch. 21 Ref. <sup>5</sup>
3	Translate the software design into computer program language (code)	R	R	G	G	<ul style="list-style-type: none"> <li>▪ Develop computer program language (code):                             <ul style="list-style-type: none"> <li>○ data structures/files,</li> <li>○ source code (where feasible),</li> <li>○ executable code</li> </ul> </li> <li>▪ Provide computer program listings</li> </ul>	<ul style="list-style-type: none"> <li>▪ Earliest practical time</li> <li>▪ Prior to final acceptance testing</li> </ul>	<ul style="list-style-type: none"> <li>▪ SD (D, R, A)</li> <li>▪ SRLM (R, A)</li> </ul>	3.3.4.b	SOFT-DESIGN, 8

<sup>1</sup> ML = Associated Management Level per Form [2033](#). R = Required. G = Required but graded.

<sup>2</sup> D = Develop or implement; R = Review; A = Approve.

<sup>3</sup> SO = Software Owner. SRLM = Software Responsible Line Manager. FDAR = Facility Design Authority Representative.

<sup>4</sup> Applicable to computer program models only.

<sup>5</sup> Ref. = [ESM](#) Chapter 21 section name and subsection number (e.g., SOFT-GEN, Subsection 3, Software Identification and Determination).

This search tool may not include the latest files; use individual files for work.

3 Section SOFT-DESIGN: Software Design

Rev. 0, 06/23/16

2.0 KEY DEFINITIONS AND ACRONYMS

See Chapter 21, SOFT-GEN for key definitions and acronyms.

3.0 SSC SOFTWARE DESIGN REQUIREMENTS (INPUT)

Define, control and verify the design. This section describes the minimum requirements, activities and deliverables for preparing SSC software design requirements (i.e., inputs). SSC software design requirements are the information provided to the SSC software designer to design the SSC software.

A. How.

1. SSC software design requirement (input) deliverables depend on the nature, complexity and associated ML of the software. Use Table 21.3-3, *SSC Software Design Requirement (Input) Deliverables* and direction from the LANL SRLM as a guide to define software design requirement deliverables.
2. Process and retain input documentation in accordance with the SRLM's governing document control and records management process. Ensure the correct SWID (obtained when completing Form 2038) is part of the documentation record number. See [AP-341-402 Engineering Document Management in Operating Facilities](#) for details.

Table 21.3-3 SSC Software Design Requirement (Input) Deliverables					
No	Deliverable	Apply by ML <sup>1</sup>			
		1	2	3	4
01	SSC System Drawings (e.g., process flow diagrams (PFDs) process and instrumentation diagrams (P&IDs), Sequence of Operations (SOOs), hardware/instrument location drawing, network drawing <sup>2</sup> , instrument/device list.)	R	R	G	G
02	SSC performance specification	R	R	G	G
03	SSC Facility Design Description (FDD) and/or System Design Description (SDD)	R	R	G	-
04	Other SSC Design inputs/outputs as required to support software design <sup>3</sup>	R	R	G	-

<sup>1</sup> R = As required by [ESM](#) Chapter 1, General, Section Z10 or other [ESM](#) chapters; G = Required but graded per [ESM](#) Chapter 1: General, Section Z10 or other [ESM](#) chapters. "-" = Not required.

<sup>2</sup> Level of detail as required by FDAR; e.g., a notional or "block" diagram of the network as a minimum, detailed technical drawing later.

<sup>3</sup> As determined by LANL SRLM; may include SSC hazard analyses, Requirements Criteria Document (RCD), identification of commercial grade dedication (CGD) critical characteristics etc.

3. SSC software design requirements (inputs) must:

- a. Be identified and documented using a systems engineering process (Ref. [P341, Facility Engineering Processes Manual](#) and [ESM](#) Chapter 20 (pending publication)).
- b. Be documented as part of the SSC design documents (e.g., Process and Instrumentation drawings (P&IDs) with Sequence of Operations (SOOs), system specifications, system design descriptions (SDDs)).
- c. Be based on upper tier performance and functional requirements.

## 3 Section SOFT-DESIGN: Software Design

Rev. 0, 06/23/16

- d. Identify the operating system, function, interfaces, performance requirements, installation considerations, applicable SSC design inputs, critical characteristics (as applicable), and design constraints of the computer program.
  - e. Specify technical and software engineering requirements, including security features.
  - f. Identify applicable reference drawings, specifications, codes, standards, regulations, procedures, or instructions that establish test, inspection, and acceptance criteria.
  - g. Be commensurate with the risk of unauthorized use; address security requirements (e.g., vulnerability/cyber-security protections).
  - h. For ML-1 through ML-3 software, be traceable throughout the software life cycle.
  - i. For ML-1 through ML-3 software, be based on system/component hazard analyses that identify system/component risks and the means for controlling them.
  - j. The software design shall consider the computer program's Operating Environment (see Definitions in SOFT-GEN).<sup>1</sup>
  - k. Identify and address the user human machine interface (HMI) requirements. Factor in existing user operational protocols, conventions and methods by users/operators. As appropriate, specify prototypes/HMI screenshots for review and comment as part of the design deliverables.
  - l. Ensure software design requirements, as applicable, are consistent with SSC technical baseline documents. For information on SSC software technical baseline documents, see [STD-342-100, Engineering Standards Manual, Chapter 1, Section Z10, General; AP-341-616, Technical Baseline Change During Design; and AP-341-405, Identification and Control of Technical Baseline, Variances, Alternate Methods, and Clarifications in Operating Facilities.](#)
  - m. Ensure the design inputs and sources are identified and documented, specified on a timely basis, translated into design documents, and their selection review and approved.
  - n. Include user-level input/review of screen shots, prototypes, etc.
4. For guidance, see:
- [IEEE Std 1233, IEEE Guide for Developing System Requirements Specifications](#), and
  - [ISO/IEC/IEEE 29148, Systems and Software Engineering-Life Cycle Processes-Requirements Engineering.](#)

**When.** Develop software design requirements at the earliest practical time and as much as practical, prior to software design. As applicable, submit software requirements (draft) at 30% design submittals, firmed up at 60%.

---

<sup>1</sup> An integral part of software design is the design of a computer program that is part of an overall system. ASME NQA-1a-2009, Part II, Subpart 2.7-402



## 3 Section SOFT-DESIGN: Software Design

Rev. 0, 06/23/16

- C. Who. For SSC software, the SSC Design Agent (or Design Agency [DA]) that develops the SSC detailed design also develops, reviews, and approves the SSC software design requirements. The LANL SRLM and FDAR review and approve the software requirements.

## 4.0 SSC SOFTWARE DESIGN (OUTPUT)

- A. How.

## 4.1 General Software Design Requirements

- A. Software design is an expanding and complex area. Accordingly, this section is limited to providing only a high-level description of the software design output requirements. For additional detail, see:
- [DOE-STD-1195](#), *Design of Safety Significant Safety Instrumented Systems Used at a DOE Nonreactor Nuclear Facilities*
  - [ANSI/ISA 84.00-01](#), *Functional Safety: Safety Instrumented Systems for the Process Industry Sector*,
  - [ANSI/IEEE STD 7-4.3.2](#), *IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations*,
  - [IEEE Std 1016](#), *IEEE Standard for Information Technology – System Design – Software Design Descriptions<sup>2</sup>*
  - [IEEE Std 1016.1](#), *IEEE Guide to Software Design Descriptions*
  - [ESM](#) Chapter 8: *Instrumentation and Controls*
- B. Design software to meet the software requirements and deliverables of this section, SOFT-DESIGN, and other ESM chapters as applicable. Design deliverables may be combined as needed in one or more documents.
- C. Use an accepted design methodology (see [IEEE Std 1016.1](#)).
- D. As specified by the LANL SRLM for ML-1 and ML-2 software, require and provide software designer and computer programmers (coders) qualifications commensurate with the risk associated with the software.
- E. Document the software design and the computational sequence necessary to meet the software requirements including as applicable: numerical methods, mathematical models, physical models, control flow, control logic, data flow, process flow, data structures, supporting calculations (Ref. [AP-341-605](#), *Calculations*), software baseline, process structures, and the applicable relationships between data structures and process structures. Combine with the documentation of the software design requirements, or the computer program listings (see definitions) resulting from implementation of the design.
- F. Ensure software design outputs, as applicable, are consistent with SSC technical baseline documents.
- G. Software design (output) deliverables depend on the nature, complexity and associated ML of the software. See SOFT-DESIGN References for examples. Deliverables consist of drawings and/or other descriptive documents. They may be combined in one or more documents or provided separately. Software design deliverables may be divided into the following categories: (a) software design

<sup>2</sup> Flow diagrams, charts and/or pseudo-code should be provided.

3 Section SOFT-DESIGN: Software Design

Rev. 0, 06/23/16

documentation (system architecture, software architecture, software detail design collectively referred to as SWDD), (b) Software Requirements Specification (SWRS), (c) hazard analysis and mitigation documentation (SWHA), (d) software design traceability documentation (SWTM), (e) O&M documentation, (f) SOFT-GEN deliverables, and (g) other software deliverables.

**Note:** Verification and Validation (V&V) documentation, including test plans, interim test reports, and design review documentation, are typically produced concurrent with the design. See SOFT-V&V.

- H. Provide associated software deliverables as required by other ESM chapters (e.g., Chapter 8 for instrumentation and control systems).
- I. Provide design deliverables using Table 21.3-4, *Software Design (Output) Deliverables* as a guide, and direction from the LANL SRLM.
- J. Ensure the design is defined, controlled and verified; the design interfaces are identified and controlled; design adequacy is verified by individuals other than those who designed the software. *Guidance: The eventual system engineer and/or users should work with the design/development team to ensure documentation etc. is done to an appropriate level commensurate with the ML of the software/SSC.*
- K. Ensure design documentation and records include not only final design documents, such as drawings and specifications, and revision to those documents, but also documentation that identifies the important steps in the design process, including sources of design inputs.

**Table 21.3-4 Software Design (Output) Deliverables**

No	Design Deliverable Element	Apply by ML <sup>1</sup>			
		1	2	3	4
01	Software system architecture design	R	R	G	G
02	Software requirements specification (SWRS)	R	R	G	-
03	Software architecture design	R	R	G	G
04	Software detail design	R	R	G	-
05	Software system hazard analysis and mitigation docs. (SWHA)	R	R	G	-
06	Software design traceability documentation (e.g., SWTM)	R	R	G	-
07	Computer Program Listings, including the program (design product)	R	R	G	-
08	Operations and maintenance (O&M) instructions	R	R	R	R
09	SOFT-GEN deliverables <sup>2</sup>	See SOFT-GEN			
10	Other software design deliverables <sup>3, 4</sup>	See LANL SRLM and other ESM chapters.			

<sup>1</sup> R = Required; G = Required but graded; "-" = Not required.  
<sup>2</sup> Includes software list, Form 2033, software data sheet, software baseline.  
<sup>3</sup> As determined by LANL SRLM and/or as required in other ESM chapters (e.g., Human Machine Interface (HMI) graphical layouts/slides). See SOFT-V&V for associated V&V deliverables.  
<sup>4</sup> See ESM Chapter 8 for software related deliverables (e.g., logic diagrams, loop diagrams, point list tables, etc.) specific to instrument and control (I&C) system designs.

- L. Retain documentation in accordance with the governing records management process. For ES-Div Non-SSC software and SSC software, ensure the correct SWID

**3 Section SOFT-DESIGN: Software Design**

Rev. 0, 06/23/16

(obtained when completing Form 2033) is part of the documentation record number. See [AP-341-402](#) for details.

**4.2 Software System Architectural Design**

- A. Develop the software system architectural design. Allocate system-level requirements to hardware, software, and interfaces of the system.
- B. For ML-1, ML-2, and some ML-3 systems where software operational complexity and lifecycle maintenance can be disadvantages, evaluate and as appropriate, employ designs that use no or minimal software.
- C. Sufficiently detail the system architectural design such that a person technically qualified in the subject can verify and validate (V&V) that the design satisfies requirements without recourse to the Software Designer (SD).
- D. Consider the following concepts, as applicable, in the software system architectural design.
  - 1. **Isolation:** Critical elements should be separated from each other to preclude undefined and/or unintended interactions. For ML-1 through ML-3 software, safety systems should be separated from non-safety systems where possible. Barriers (e.g., separation of safety from non-safety modules) should be used to prevent non-safety functions from interfering with safety functions. Consider use of different platforms, power supplies, inputs/outputs to physically separate the safety from non-safety elements.
  - 2. **Independence/Diversity:** For ML-1 and ML-2 software, independent, diverse systems should be employed. These are systems where the stimuli originate from and are handled by separate elements with different designs, independent hardware inputs and/or independent software modules.

**Note:** Because different manufacturers may use the same processor or software, common mode failures may not be averted simply by acquiring from several manufacturers.
  - 3. **Fail-Safe Design:** Design such that the system remains in a safe mode upon failure without compromising isolation features or other safety functions.
  - 4. **Incompatibility/Longevity:** Design to ensure integrated hardware-software compatibility as much as possible throughout the life-cycle of the system.

**4.3 Software Requirements Specification (SWRS)**

- A. For software system requirements that are allocated to software, further refine the requirements and develop software-level requirements. Document the software requirements. Software requirements are documented in a software requirements specification (SWRS). See [IEEE Std 830](#), *IEEE Recommended Practice for Software Specifications* for guidance.

**4.4 Software Architectural Design**

- A. Transform the software requirements into the software architectural design. The software architecture design specifies the structures of the software and the various software components, interactions and interfaces between the software components.

**3 Section SOFT-DESIGN: Software Design**

Rev. 0, 06/23/16

- B. Include the high-level design for the external interfaces between the software components and other components within the system (hardware and human interface) and between the software and entities outside the system.
- C. Sufficiently detail the software architectural design such that a person technically qualified in the subject can V&V the design without recourse to the SD.
- D. Where appropriate and/or as required by the software requirements document, design software control functions that are performed incrementally rather than in a single step to reduce the potential that a single failure of a software element can cause an unsafe system state.
- E. Where appropriate and/or as required by the software requirements document, design built-in fault detection and self-diagnostics that detect and report software faults and failures in a timely manner and allow actions to be taken to avoid an impact on the system's safe operation.

**4.5 Software Detail Design**

- A. For ML-1 through ML-3 software, define the internals of each software component down to individual software modules that can be coded.
- B. Specify/document the software coding standards and/or conventions.
- C. As applicable, include data structures/databases and external interfaces.
- D. Sufficiently detail the detailed software design to: (a) allow a person technically qualified in the subject to V&V the design without recourse to the SD; and (b) allow computer programmers to code the design without undue difficulty.

**4.6 Software System Hazard Analysis and Mitigation**

- A. For ML-1 through ML-3 software, conduct and document a **software** hazard analysis at the system and component level to identify software risks and develop mitigating approaches for controlling them. Potential failures should be identified and evaluated for their consequences of failure and probability of occurrence. Some potential problems may include (1) complex or faulty algorithm, (2) lack of proper handling of incorrect data or error conditions, (3) buffer overflow, and (4) incorrect sequence of operations due to either logic or timing faults.
- B. Ensure the software hazard analysis is consistent with the system safety documentation for the associated facility. See [SBP111-1, Facility Hazard Categorization and Documentation](#) for associated facility safety documentation.
- C. For ML-1 and ML-2 software, perform and document the hazard analysis based on recognized consensus standards. See ESM [Chapter 8, Instrument & Controls](#) for application of hazard analysis/mitigation with respect to the following standards for SSG software:
  - ANSI/Instrumentation, Systems, and Automation Society ([ISA S84.00.01, Functional Safety: Safety Instrumented Systems for the Process Industry Sector](#)), and,
  - [ANSI/IEEE Std 7-4.3.2](#), IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations).
- D. For ML-3 software, standard-based methods or less formal methods should be used using a graded approach (e.g., failure modes and effects analysis, fault-tree modeling, event-tree modeling, cause-consequence diagrams, hazard and operability analysis, and interface analysis).

**3 Section SOFT-DESIGN: Software Design**

Rev. 0, 06/23/16

- E. For ML-4 software, a hazard analysis is not required but may be performed.
- F. *Multiple/common-cause failures should be evaluated in the hazard analysis.*  
**Note:** Failure mode and effect analysis (FMEA) approaches, when used alone, do not address multiple failures/common-cause failures. ([ANSI/ISA 84.00-01-2004-Part 1](#) and [IEEE STD-7-4.3.2-2003](#) provide guidance.)
- G. The hazard analysis and design must include analysis and possible problems with the computer program's operating environment (including security environment) and external and internal abnormal conditions and events that can affect the computer program.
- H. In the software design documentation (SWDD), as part of the SWHA, or in a separate deliverable, provide documentation that shows how the consequences of hazards/problems are mitigated. *Mitigation strategies should be included for:*
  1. *Software standard hazards (i.e., the basic hazards associated with the software or process);*
  2. *Software system failures (i.e., the functionality written into the software itself to protect from failure); and*
  3. *Software system overrides (i.e., the functionality written into the software to keep a user or other system from bypassing the safety features within the software item.)*

**4.7 Software Design Traceability and Testability**

- A. For ML-1 and ML-2 software, document traceability from the software design requirements to the software design deliverables. For ML-3 software, a software traceability matrix (SWTM) is recommended to document traceability. Other methods however, such as tracing the requirement within the design deliverable are also acceptable. Requirements tools and/or databases may be used.
- B. For large, complex ML-1 and ML-2 software, consider bi-directional traceability (tracing from the requirements to the design as well as from the design to the requirements) to identify functionality that may have been inadvertently added without requirement drivers.
- C. Design the software and the associated SSC such that it can readily be tested against the requirements.

**4.8 Operations and Maintenance (O&M) Instructions**

- A. Develop and provide operations and maintenance (O&M) instructions to use and maintain the computer program(s).
- B. Sufficiently detail the O&M instructions to allow an individual trained in the use of such software to follow the instructions and use the program without undue difficulty and/or probability of misuse.
- C. *For ML-1 and ML-2 software, O&M instructions should be developed using standard conventions for instructions (e.g., [ANSI/IEEE Std 26514](#), Systems and Software Engineering – Requirements for Designers and Developers of User Documentation).*

**4.9 SOFT-GEN Requirements and Deliverables**

Satisfy software requirements and provide deliverables as described in SOFT-GEN.

**3 Section SOFT-DESIGN: Software Design**

Rev. 0, 06/23/16

- A. When. Design software at the earliest practical time and as much as practical, prior to software coding. Unless the software design is deferred to subcontractors, submit software design deliverables with the 90% and 100% detail designs. If software design is deferred, then specify and submit software design deliverables as part of the required submittal process.
- B. Who. The SD develops, reviews and approves the deliverables of this section including the architectural design, software specification, detail design, hazard analysis and mitigation documentation, and design traceability documentation. The SD, SRLM (prior to turnover to LANL), and LANL SRLM review and approve.

**5.0 SSC SOFTWARE COMPUTER PROGRAM LANGUAGE (CODE)**

- A. How.
1. Translate the software design into computer program language (code).
  2. For ML-1 through ML-3 software, use standards and/or conventions as approved in the software design (e.g., [IEEE Std 1066](#), *IEEE Standard for Standard System C Language Reference*).
  3. Develop source code suitable for compilation or translation.
  4. Provide both source code and/or executable code as deliverables. If source code cannot be provided (e.g., for proprietary reasons), then provide objective evidence that the source code was V&V'd as per [IEEE Std. 1012](#) and/or acceptable methods.
  5. For ML-1 and ML-2 software, provide computer program listings (see definitions).
  6. Code should be developed to sufficient detail and clarity to allow someone technically qualified in the computer programming language to review, understand, and verify it meets the software design requirements without recourse to the SD and/or coder.
  7. For ML-1 and ML-2 code, ensure most code lines have explanatory notes/comments (pseudo code) to support future troubleshooting, bug fixes, and/or system modifications.
  8. Employ configuration management (including computer program code labeling) and problem reporting/corrective management in accordance with SOFT-GEN.
- B. When. Develop code at the earliest practical time and prior to final program acceptance testing.
- C. Who. The SD develops, reviews and approves the software code; the SRLM reviews and approves the code.

**6.0 NON-SSC SOFTWARE DESIGN REQUIREMENTS (INPUT)**

*Guidance: Unlike SSC software, Non-SSC software typically does not use upper level SSC design documents (e.g., P&IDs, SOOs, SDDs) for design input. Therefore, the design input is communicated through other documentation. A **system** requirements specification (SWYRS) is used to convey upper level design inputs analogous to the upper-level SSC design documents. A software requirements specification (SWRS) is developed for software system requirements that are allocated to software. The SWRS further refines the requirements and develops software-level requirements. For designs where software is*

## 3 Section SOFT-DESIGN: Software Design

Rev. 0, 06/23/16

relatively low in complexity and when allowed by the LANL SRLM, the SWYRS and SWRS may be combined.

A. How.

1. Using a graded approach for ML1 through ML-4 associated software, develop a system requirements specification (SWYRS). The SWYRS is a document that communicates the requirements of the customer to the technical community who will further specify and build the software. See the following for SWYRS guidance:
  - [ISO/IEC/IEEE 29148](#), *Systems and Software Engineering-Life Cycle Processes-Requirements Engineering*.
2. The SWYRS must:
  - a. Be identified and documented using a systems engineering process (Ref. [P341](#), *Facility Engineering Processes Manual* and [ESM](#) Chapter 20 [pending publication]).
  - b. Identify the operating system and other aspects of Operating Environment, function, interfaces, performance requirements, installation considerations, applicable design inputs, and design constraints of the computer program.
  - c. Specify technical and software engineering requirements, including security features.
  - d. Identify applicable reference drawings, specifications, codes, standards, regulations, procedures, or instructions that establish test, inspection, and acceptance criteria.
  - e. Be commensurate with the risk of unauthorized use; address security requirements (e.g., vulnerability/cyber-security protections).
  - f. For ML-1 through ML-3 software, be traceable throughout the software life cycle.
  - g. For ML-1 through ML-3 software, be based on system/component hazard analysis that identifies system/component risks and the means for controlling them.

B. When. Develop the SWYRS at the earliest practical time and where practical, prior to software design.C. Who. The SO develops the SWYRS and the SO (e.g., with assistance of SD); SRLM and FDAR review and approve the SWYRS.

## 3 Section SOFT-DESIGN: Software Design

Rev. 0, 06/23/16

## 7.0 NON-SSC SOFTWARE DESIGN (OUTPUT)

A. How.

1. See SSC Software design Subsection 4 and note that SSC-related text and/or references do not apply. For example, the following are not applicable:
  - a. Integration of design outputs with SSC technical baseline documents (4.1.F)
  - b. Provide associated software deliverables or use associated software standards as required for SSCs as stated by other FSM chapters (4.1.B and 4.6.C).
2. *Non-SSC software can include computer models. Models are simplifications of the real world constructed to gain insights into select attributes of a particular physical, biological, economic, engineered, or social system.*<sup>3</sup> For computer model design, also include the following:
  - a. A description of the conceptual model for the problem to be solved. Include assumptions, algorithms, relationships and data.
  - b. The mathematical terms that describe the conceptual model. Include mathematical equations, boundary values, initial conditions, and modeling data as required. Include numerical solution techniques such as finite element and finite difference when equations describing the mathematical model cannot be solved analytically.
  - c. *Guidance: Use [ANSI/ANS-10.7](#), Non-Real-Time, High-Integrity Software for the Nuclear Industry – Development Requirements as a guide.*

B. When. Design software at the earliest practical time and as much as practical, prior to software coding.

C. Who. The SD develops, reviews, and approves the software design; the SO and SRLM review and approve the design.

## 8.0 NON-SSC SOFTWARE PROGRAM LANGUAGE (CODE)

See Subsection 5.

## REFERENCE: SOFTWARE DESIGN DOCUMENTATION EXAMPLES

Examples initially provided are for informational purposes only and have not been reviewed for compliance to the requirements of this chapter. Better examples will be posted as they become available.

<sup>3</sup> [EPA/100/K-09/003](#), *Guidance on the Development, Evaluation, and Application of Environmental Models*.



**TABLE OF CONTENTS**

1.0 PURPOSE, SCOPE AND APPLICABILITY ..... 2

2.0 KEY DEFINITIONS AND ACROYNMS..... 9

3.0 V&V AND SOFTWARE APPROVAL FOR USE (SWAU) - GENERAL ..... 9

4.0 DESIGNED SSC SOFTWARE V&V and SWAU ..... 11

    4.1 Design Requirement (Input) Review .....12

    4.2 Test Plan Development and Review ..... 12

    4.3 Software Design Review ..... 14

    4.4 Computer Program Code (Code) Review ..... 15

    4.5 Testing/Cx, Review and Report ..... 15

    4.6 Review and Approval for Use..... 17

5.0 READ-ONLY SSC SOFTWARE V&V AND SWAU ..... 18

    5.1 Requirement (Input) Review ..... 18

    5.2 Test Plan Development and Review ..... 18

    5.3 Design Review ..... 19

    5.4 Testing/Cx and Review ..... 19

    5.5 Review and Approval for Use..... 19

6.0 DESIGNED NON-SSC SOFTWARE V&V AND SWAU ..... 19

    6.1 Model V&V ..... 20

    6.2 Design Requirement (Input) Review ..... 20

    6.3 Test Plan Development and Review ..... 20

    6.4 Software Design Review ..... 20

    6.5 Designed Non-SSC Computer Program Code Review ..... 20

    6.6 Testing and Review..... 20

    6.7 Review and Approval for Use..... 20

7.0 ACQUIRED NON-SSC SOFTWARE V&V AND SWAU ..... 21

    7.1 Model V&V ..... 21

    7.2 Acquisition Requirement (Input) Review ..... 21

    7.3 Test Plan Development and Review ..... 21

    7.4 Acquisition Review ..... 22

    7.5 Testing and Review..... 22

    7.6 Review and Approval for Use..... 22

8.0 TOOLBOX CODE V&V AND SWAU..... 22

9.0 SIMPLE & EASILY UNDERSTOOD NON-SSC SOFTWARE V&V/SWAU ..... 22

APPENDICES, FORMS AND REFERENCES..... 23

APPENDICES: ..... 23

APPENDIX A: COMPUTER PROGRAM TEST PLAN REQUIREMENTS ..... 24

FORMS:

    SOFT-V&V-FM01: SOFTWARE APPROVAL FOR USE FORM (SWAU)

REFERENCES:

    EXAMPLE REQUIREMENTS TRACEABILTIY MATRIX

    EXAMPLE TEST PLAN - NO 1.

    EXAMPLE TEST REPORT - NO. 1

    EXAMPLE TEST REPORT - NO. 2

Rev	Date	Description	POC	RM
0	06/23/16	Initial issue	Tobin Oruch, ES-DO	Mel Burnett, CENG-OFF

## 1.0 PURPOSE, SCOPE, AND APPLICABILITY

This section describes the “how, when, and who” for:

- verifying and validating (V&V) software, and
- approving software for use (SWAU)

See Tables 21.4-1 and 21.4-2 for a summary of SOFT-V&V for SSC software and Non-SSC software respectively.

See Chapter 21 applicability in SOFT-GEN. In addition:

1. For software changes during V&V, see SOFT-GEN.
2. Use this section for V&V and SWAU of software that is acquired and for software that is designed.
3. Read-Only: This section includes V&V for Read-Only SSC software. This is SSC software where the computer program cannot be modified other than through replacement of the computer program or the associated SSC. It is sometimes referred to as embedded software or firmware; however, for this section, the term Read-Only SSC software is used.
4. For SSC software, use this section in conjunction with [ESM](#) Chapter 15, *Commissioning*.
5. For Non-SSC software, this section includes V&V and SWAU for: (a) toolbox codes and (b) software used in SSC design that is Simple and Easily Understood. See SOFT-GEN for definitions.
6. For computer programs that are commercially dedicated, the applicable V&V requirements of this section are required for critical characteristics when specified in the CGD technical evaluation. SWAU requirements of this chapter also apply.

This search tool may not include the latest files; use individual files for work.

4 Section SOFT-V&V: Software Verification and Validation

Rev. 0, 06/23/16

**Table 21.4-1 Software SOFT-V&V Summary for SSC Software**  
 (This table is a summary only and does not include all requirement details. See text for details.)

Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			P1040 R9 Ref. Section	ESM Ch. 21 Ref. <sup>5</sup>
		1	2	3	4	How	When <sup>2</sup>	Who <sup>3,4</sup>		
<b>1.0 For Designed SSC Software</b>										
1	Review design (input) requirements	R	R	G	G	<ul style="list-style-type: none"> <li>Review design input documents (docs.) to SOFT-DESIGN design (input) requirement criteria</li> <li>As applicable, ensure Commercial Grade Dedication (CGD) requirements are addressed</li> </ul>	<ul style="list-style-type: none"> <li>Prior to software design as much as practical</li> </ul>	<ul style="list-style-type: none"> <li>SD ( R)</li> <li>SRLM (R)</li> <li>LCxA (R)</li> <li>FDAR (R)</li> </ul>	3.3.4; 3.3.5	SOFT-V&V, 4; SOFT-DESIGN, 3
2	Develop/ review test plan (SWTP)	R	R	G	G	<ul style="list-style-type: none"> <li>Develop software test plan (SWTP) and submit 60%, 90% and 100% test plan submittals</li> <li>Review test plan to SOFT-V&amp;V test plan criteria</li> <li>Integrate into/with SSC Commissioning (Cx) docs.</li> </ul>	<ul style="list-style-type: none"> <li>After requirements and concurrent with design</li> <li>Prior to interim testing and/or acceptance testing</li> </ul>	<ul style="list-style-type: none"> <li>SD (D, R, A)</li> <li>LCxA (R)</li> <li>SRLM (R, A)</li> </ul>	3.3.4	SOFT-V&V, 4
3	Review software design (SWDD)	R	R	G	G	<ul style="list-style-type: none"> <li>Review to SOFT-DESIGN and SOFT-V&amp;V criteria</li> <li>Review 60%, 90% and 100% software design documents/submittals</li> </ul>	<ul style="list-style-type: none"> <li>Per software design schedule</li> <li>Prior to acceptance testing</li> </ul>	<ul style="list-style-type: none"> <li>SD (R)</li> <li>SRLM (R)</li> <li>IR (R)</li> </ul>	3.3.4	SOFT-V&V, 4; SOFT-DESIGN, 3,4
4	Review computer program code	R	R	-	-	<ul style="list-style-type: none"> <li>Review computer program code where feasible</li> </ul>	<ul style="list-style-type: none"> <li>Prior to acceptance testing</li> </ul>	<ul style="list-style-type: none"> <li>SD (R)</li> <li>SRLM (R)</li> <li>IR (R)</li> </ul>	3.3.4	SOFT-V&V, 4
5	Testing/Cx & review	R	R	G	G	<ul style="list-style-type: none"> <li>Test (including interim tests) per SSC design, software test plan, and Cx docs.</li> <li>Perform acceptance test in simulated test environment (i.e., test bed)</li> <li>Perform acceptance test in operating environment</li> <li>Provide, review &amp; approve test (qualification) report (SWTR)</li> </ul>	<ul style="list-style-type: none"> <li>Per software test plan, Cx docs, and SSC design</li> <li>Prior to approval for use</li> </ul>	<ul style="list-style-type: none"> <li>T (D, R, A)</li> <li>SD (R, A)</li> <li>SO (R, A)</li> <li>LCxA, (R, A)</li> <li>SRLM (R, A)</li> </ul>	3.3.5	SOFT-V&V, 4
6	Review/ SWAU	R	R	G		<ul style="list-style-type: none"> <li>Prepare software approval for use (SWAU) request package</li> <li>Review to SOFT-MAINT SWAU criteria</li> <li>Approve and document SWAU</li> </ul>	<ul style="list-style-type: none"> <li>Prior to intended use</li> </ul>	<ul style="list-style-type: none"> <li>SO (D)</li> <li>SRLM (R, A)</li> </ul>	3.3.5; 3.3.6	SOFT-V&V, 4; SOFT-MAINT, 3

This search tool may not include the latest files; use individual files for work.

4 Section SOFT-V&V: Software Verification and Validation

Rev. 0, 06/23/16

**Table 21.4-1 Software SOFT-V&V Summary for SSC Software**  
 (This table is a summary only and does not include all requirement details. See text for details.)

Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			P1040 R9 Ref. Section	ESM Ch. 21 Ref. <sup>5</sup>
		1	2	3	4	How	When <sup>2</sup>	Who <sup>3,4</sup>		
<b>2.0 For Read Only SSC Software</b>										
1	Review (Input) requirements	R	R	G	G	<ul style="list-style-type: none"> <li>Review SSC design/acquisition requirement (input) docs. (e.g. Statement of Work, [SOW]) per governing doc. control process</li> <li>As applicable, ensure CGD requirements are addressed</li> </ul>	<ul style="list-style-type: none"> <li>Prior to acquisition</li> </ul>	<ul style="list-style-type: none"> <li>Per governing doc. control process</li> </ul>	3.3.4; 3.3.5	SOFT-V&V,5; SOFT-DESIGN, 3; SOFT-ACQUIRE, 3
2	Develop and review test plan	R	R	G	G	<ul style="list-style-type: none"> <li>Use SSC Cx documents per <a href="#">ESM</a> Chapter 15 as software test plan documents</li> </ul>	<ul style="list-style-type: none"> <li>Per <a href="#">ESM</a> Chapter 15</li> <li>Prior to Cx</li> </ul>	<ul style="list-style-type: none"> <li>Per <a href="#">ESM</a> Chapter 15</li> </ul>	3.3.4	SOFT-V&V, 5
3	Review the design	R	R	G	G	<ul style="list-style-type: none"> <li>Review software as part of the SSC design deliverables (<a href="#">ESM</a> Chapter 15)</li> </ul>	<ul style="list-style-type: none"> <li>Per SSC design schedule</li> </ul>	<ul style="list-style-type: none"> <li>Per governing SSC review process</li> </ul>	3.3.4	SOFT-V&V, 5
4	Cx and review	R	R	G	G	<ul style="list-style-type: none"> <li>Test per SSC design and Cx docs.</li> <li>Provide, review and approve Cx report per <a href="#">ESM</a> Chapter 15</li> </ul>	<ul style="list-style-type: none"> <li>Per Cx docs. and SSC design</li> <li>Prior to approval for use</li> </ul>	<ul style="list-style-type: none"> <li>Per <a href="#">ESM</a> Chapter 15</li> </ul>	3.3.5	SOFT-V&V, 5;
5	Review and approve for use	R	R	G	-	<ul style="list-style-type: none"> <li>Prepare SWAU request package</li> <li>Review to SOFT-MAINT SWAU criteria</li> <li>Approve and document SWAU</li> </ul>	<ul style="list-style-type: none"> <li>Prior to intended use</li> </ul>	<ul style="list-style-type: none"> <li>SO (D)</li> <li>SRLM (R, A)</li> </ul>	3.3.5; 3.3.6	SOFT-V&V, 5; SOFT-MAINT, 3

<sup>1</sup> ML = Management Level; R = Required; G = Required but graded. "-" = Not required.

<sup>2</sup> "When" constitutes the V&V control points (points in the process where reviews are performed).

<sup>3</sup> D = Develop or Execute; R = Review; A = Approve.

<sup>4</sup> SD = Software Designer; SO = Software Owner. SRLM = Software Responsible Line Manager. FDAR = Facility Design Authority Representative; T = Testers; IR = Independent Reviewer. LCxA = LANL Commissioning Authority. See associated [ESM](#) Chapter 21, SOFT-GEN for more information.

<sup>5</sup> Ref. = [ESM](#) Chapter 21 section name and subsection number (e.g., section SOFT-GEN, Subsection 3, Software Identification and Determination).

This search tool may not include the latest files; use individual files for work.

4 Section SOFT-V&V: Software Verification and Validation

Rev. 0, 06/23/16

Table 21.4-2 SOFT-V&V Summary for <u>Non-SSC</u> Software (This table is a summary only and does not include all requirement details. See text for details.)										
Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			P1040 R9 Ref. Section	ESM Ch. 21 Ref. <sup>6</sup>
		1	2	3	4	How	When <sup>2</sup>	Who <sup>3,4</sup>		
<b>1.0 For <u>Designed</u> Non-SSC Software</b>										
1	Review design (input) requirements	R	R	R	R	<ul style="list-style-type: none"> <li>Review design input docs. to SOFT-DESIGN design (input) requirement criteria</li> <li>As applicable, ensure Commercial Grade Dedication (CGD) requirements are addressed</li> </ul>	<ul style="list-style-type: none"> <li>Prior to software design as much as practical</li> <li>Prior to software acquisition</li> </ul>	<ul style="list-style-type: none"> <li>SD (R)</li> <li>SRLM (R)</li> </ul>	3.3.4; 3.3.5	SOFT-V&V, 6; SOFT-DESIGN, 6
2	Model V&V	R	R	G	G	<ul style="list-style-type: none"> <li>As applicable, V&amp;V the appropriateness of the model in addition to the computer code</li> <li>Document in design and testing reviews</li> </ul>	<ul style="list-style-type: none"> <li>After requirements and concurrent with design</li> </ul>	<ul style="list-style-type: none"> <li>SRLM (R, A)</li> </ul>	3.3.4; 3.3.5	SOFT-V&V, 6; SOFT-DESIGN, 6
3	Develop/review test plan	R	R	G	-	<ul style="list-style-type: none"> <li>Develop software test plan (SWTP) and submit 60%, 90% and 100% test plan submittals</li> <li>Review test plan to SOFT-V&amp;V test plan criteria</li> </ul>	<ul style="list-style-type: none"> <li>After requirements and concurrent with design</li> <li>Prior to formal interim testing and/or acceptance testing</li> </ul>	<ul style="list-style-type: none"> <li>SD (D, R, A)</li> <li>SRLM (R, A)</li> </ul>	3.3.4	SOFT-V&V, 6
4	Review software design	R	R	G	G	<ul style="list-style-type: none"> <li>Review to SOFT-DESIGN design and SOFT-V&amp;V criteria</li> <li>Review 60%, 90% and 100% software design documents/submittals</li> </ul>	<ul style="list-style-type: none"> <li>Per software design schedule</li> <li>Prior to acceptance testing</li> </ul>	<ul style="list-style-type: none"> <li>SD (R)</li> <li>SRLM (R)</li> <li>IR (R)</li> </ul>	3.3.4	SOFT-V&V, 6; SOFT-DESIGN, 6, 7
5	Review computer program code	R	R	-	-	<ul style="list-style-type: none"> <li>Review computer program code where feasible</li> </ul>	<ul style="list-style-type: none"> <li>Prior to acceptance testing</li> </ul>	<ul style="list-style-type: none"> <li>SD (R)</li> <li>SRLM (R)</li> <li>IR (R)</li> </ul>	3.3.4	SOFT-V&V, 6, 8

This search tool may not include the latest files; use individual files for work.

4 Section SOFT-V&V: Software Verification and Validation

Rev. 0, 06/23/16

Table 21.4-2 SOFT-V&V Summary for <u>Non-SSC</u> Software (This table is a summary only and does not include all requirement details. See text for details.)										
Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			P1040 R9 Ref. Section	ESM Ch. 21 Ref. <sup>6</sup>
		1	2	3	4	How	When <sup>2,3</sup>	Who <sup>3,4</sup>		
6	Testing and review	R	R	G	G	<ul style="list-style-type: none"> <li>Test per software test plan</li> <li>Provide, review and approve test report (SWTR)</li> </ul>	<ul style="list-style-type: none"> <li>Per software test plan,</li> <li>Prior to approval for use</li> </ul>	<ul style="list-style-type: none"> <li>T (D, R, A)</li> <li>SD (R, A)</li> <li>SO (R, A)</li> <li>SRLM (R, A)</li> </ul>	3.3.5	SOFT-V&V, 6
7	Review/ SWAU	R	R	G	-	<ul style="list-style-type: none"> <li>Prepare SWAU request package</li> <li>Review to SOFT-MAINT SWAU criteria</li> <li>Approve and document SWAU</li> </ul>	<ul style="list-style-type: none"> <li>Prior to intended use</li> </ul>	<ul style="list-style-type: none"> <li>SO (D)</li> <li>SRLM (R, A)</li> </ul>	3.3.5; 3.3.6	SOFT-V&V, 6; SOFT-MAINT, 4
<b>2.0 For <u>Acquired</u> Non-SSC Software</b>										
1	Model V&V	R	R	G	G	<ul style="list-style-type: none"> <li>As applicable, V&amp;V the appropriateness of the model in addition to the computer code</li> <li>Document in acquisition and testing reviews</li> </ul>	<ul style="list-style-type: none"> <li>Prior to SWAU</li> </ul>	<ul style="list-style-type: none"> <li>SRLM (R, A)</li> </ul>	3.3.4; 3.3.5	SOFT-V&V, 7; SOFT-DESIGN, 7 SOFT-ACQUIRE, 4
2	Review acquisition (input) requirements	R	R	R	R	<ul style="list-style-type: none"> <li>Review acquisition input docs. (e.g., SOW) to SOFT-ACQUIRE acquisition (input) requirement criteria</li> <li>As applicable, ensure CGD requirements are addressed</li> </ul>	<ul style="list-style-type: none"> <li>Prior to software acquisition</li> </ul>	<ul style="list-style-type: none"> <li>SO ( R)</li> <li>SRLM (R)</li> </ul>	3.3.4; 3.3.5	SOFT-V&V, 7; SOFT-ACQUIRE, 4
3	Develop/review test plan	R	R	G		<ul style="list-style-type: none"> <li>Develop software test plan (SWTP)</li> <li>Review test plan to SOFT-V&amp;V test plan criteria</li> </ul>	<ul style="list-style-type: none"> <li>After requirements and concurrent with design</li> <li>Prior to acceptance testing</li> </ul>	<ul style="list-style-type: none"> <li>SD (D, R, A)</li> <li>SRLM (R, A)</li> </ul>	3.3.4	SOFT-V&V, 7

This search tool may not include the latest files; use individual files for work.

Table 21.4-2 SOFT-V&V Summary for <u>Non-SSC</u> Software (This table is a summary only and does not include all requirement details. See text for details.)										
Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			P1040 R9 Ref. Section	ESM Ch. 21 Ref. <sup>6</sup>
		1	2	3	4	How	When <sup>2,3</sup>	Who <sup>3,4</sup>		
4	Review acquisition	R	R	G	G	<ul style="list-style-type: none"> <li>Review to conformance with acquisition documents</li> </ul>	<ul style="list-style-type: none"> <li>Prior to acceptance testing</li> </ul>	<ul style="list-style-type: none"> <li>SO (R)</li> </ul>	3.3.4	SOFT-V&V, 7
5	Testing and review	R	R	G	G	<ul style="list-style-type: none"> <li>Test per software test plan</li> <li>Provide, review and approve test report (SWTR)</li> </ul>	<ul style="list-style-type: none"> <li>Per software test plan,</li> <li>Prior approval for use</li> </ul>	<ul style="list-style-type: none"> <li>T (D, R, A)</li> <li>SD (R, A)</li> <li>SO (R, A)</li> <li>SRLM (R, A)</li> </ul>	3.3.5	SOFT-V&V, 7
6	Review/ SWAU	R	R	G	-	<ul style="list-style-type: none"> <li>Prepare SWAU request package</li> <li>Review to SOFT-MAINT SWAU criteria</li> <li>Approve and document SWAU</li> </ul>	<ul style="list-style-type: none"> <li>Prior to intended use</li> </ul>	<ul style="list-style-type: none"> <li>SO (D)</li> <li>SRLM (R, A)</li> </ul>	3.3.5; 3.3.6	SOFT-V&V, 7; SOFT-MAINT, 4
<b>3.0 For Acquired Non-SSC (OTS) – <u>Toolbox Codes</u></b>										
1	V&V and SWAU	R	R	R	R	<ul style="list-style-type: none"> <li>V&amp;V per instructions for the toolbox code</li> <li>Prepare SWAU request package</li> <li>Review to SOFT-MAINT SWAU criteria</li> <li>Approve and document SWAU</li> </ul>	<ul style="list-style-type: none"> <li>For each toolbox code installation</li> <li>Prior to intended use</li> </ul>	<ul style="list-style-type: none"> <li>SO (D, R, A)</li> <li>SRLM (R, A)</li> </ul>	2.2.3	SOFT-V&V, 8; SOFT-MAINT, 4

This search tool may not include the latest files; use individual files for work.

4 Section SOFT-V&V: Software Verification and Validation

Rev. 0, 06/23/16

Table 21.4-2 SOFT-V&V Summary for <u>Non-SSC</u> Software (This table is a summary only and does not include all requirement details. See text for details.)										
Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			P1040 R9 Ref. Section	ESM Ch. 21 Ref. <sup>6</sup>
		1	2	3	4	How	When <sup>2</sup>	Who <sup>3, 4</sup>		
4.0 For <u>Simple &amp; Easily Understood</u> Non-SSC Software Used in Design of SSCs										
1	V&V as part of SSC design V&V	R	R	G	G	<ul style="list-style-type: none"> <li>V&amp;V using the methods previously described for Non-SSC software, or, V&amp;V the software each time it is used for SSC design in accordance with the SSC design process</li> <li>Provide and review minimum software information (e.g., computer-generated evidence of the programmed algorithms or equations)</li> <li>Apply appropriate SSC design review rigor to ensure the computer program produces satisfactory results.</li> </ul>	<ul style="list-style-type: none"> <li>Each time the software is used for an SSC design and when required by the SSC design V&amp;V process</li> <li>Prior to intended use</li> </ul>	<ul style="list-style-type: none"> <li>As per the applicable SSC design V&amp;V process</li> </ul>	2.2.4	SOFT-V&V, 9; SOFT-MAINT, 4
<p><sup>1</sup> ML = Management Level; R = Required; G = Required but graded.</p> <p><sup>2</sup> "When" constitutes the V&amp;V control points (points in the process where V&amp;V activities are performed).</p> <p><sup>3</sup> D = Develop or Execute; R = Review or Inspect; A = Approve.</p> <p><sup>4</sup> SD = Software Designer; SO = Software Owner. SRLM = Software Responsible Line Manager. FDAR = Facility Design Authority Representative; T = Testers; IR = Independent Reviewer.</p> <p><sup>5</sup> A SD subject matter expert (SME) other than the SD that developed the code shall be used to perform the code review.</p> <p><sup>6</sup> Ref. = <a href="#">ESM</a> Chapter 21 section name and subsection number (e.g., Section SOFT-GEN, Subsection 3, Software Identification and Determination).</p>										

This search tool may not include the latest files; use individual files for work.



## 2.0 KEY DEFINITIONS AND ACRONYMS

See SOFT-GEN for definitions and acronyms and note the following: Often the terms verification and validation are confused with each other. In simplified layman's terms: verification indicates whether "you built the item correctly" whereas validation indicates whether "you built the right item."

## 3.0 V&V AND SOFTWARE APPROVAL FOR USE (SWAU) - GENERAL

The following requirements apply to both SSC and Non-SSC software:

- A. Base the extent of V&V on the complexity of the software, the degree of standardization, the similarity with previously approved software, and the importance to safety.

Note: For the following Paras B through E, the documentation details are specified in subsequent subsections.

- B. For ML-1 and ML-2 software, trace, document, and V&V the software from requirements through design, acquisition, testing and SWAU. Refer to [ESM Chapter 20, Systems Engineering](#) (future). See *SOFT-V&V References: V&V Examples* for an example software requirements matrix (SWTM).
- C. For ML-1 through ML-3 software, document the V&V results, including the identification of the verifier/validator.
- D. For ML-1 and ML-2 reviews, document the reviewers, reviewer responsibilities, comments, and dispositions; retain reviewer comments and dispositions as records per the record retention schedule but at least until the software is approved for use.
- E. For ML-3 and ML-4 reviews, at a minimum, document the reviewers (names), the reviewer comments and retain the comments as records.
- F. Ensure the limits of use of the V&V are clearly defined; if use is desired outside of the original V&V, evaluate and as appropriate, V&V to incorporate extended functionality.
- G. For other than acceptance testing reviews, if the review alone is inadequate to determine if requirements are satisfied, use alternate calculations and/or develop and integrate (interim) tests to support the review.
- H. Suppliers that design software that is then acquired by LANL must V&V software based on Subsection 4 and Subsection 6 respectively. Perform as part of the supplier's software product development. For ML-1 and ML-2 software, suppliers must provide V&V processes and objective evidence of the supplier's V&V for LANL review.
- I. LANL workers must complete reviews as per this section and the governing document control and records management process. For ES-Div Non-SSC software and SSC software, ensure the correct SWID (obtained when completing Form [2033](#)) is part of the documentation record number. See [AP-341-402, Engineering Document Management in Operating Facilities](#) for details.  

Protect against suspect/counterfeit items (S/CI) items. As applicable, review and ensure valid software licenses. See [P330-9, Suspect Counterfeit Items \(S/CI\)](#) for additional information on identifying and processing an S/CI item.
- K. Software design verification shall be performed by competent individual(s) or group(s) other than those who developed and documented the original design, but

## 4 Section SOFT-V&amp;V: Software Verification and Validation

Rev. 0, 06/23/16

who may be from the same organization. cursory supervisory reviews do not satisfy the intent of this chapter. Verification may be performed by the originator's supervisor, provided (a) the supervisor did not specify a singular design approach or rule out certain design considerations and did not establish the design inputs used in the design, or (b) the supervisor is the only individual in the organization competent to perform the verification.

- L. Verification must include any one or a combination of design reviews, alternate calculations, and tests performed during computer program development. Although multiple reviews are specified herein, the reviews may be performed and documented separately or combined, as appropriate.
- M. Software tools must be evaluated, reviewed, tested, accepted for use, and placed under configuration control; software tools that do not affect the performance of the software need not be placed under configuration control.
- N. When commercial grade dedication (CGD or dedication) is used for V&V software, satisfy NQA-1, Part II, Subpart 2.14, *Quality Assurance Requirements for Commercial Grade Items and Services* and the following:
1. The dedication process shall be documented and include:
    - a. Identification of the capabilities and limitations for intended use as critical characteristics,
    - b. Utilization of test plans and test cases as the method of acceptance to demonstrate the capabilities within the limitations, and
    - c. Instructions for use (e.g. user manual) within the limits of the dedicated capabilities.
  2. The performance of the actions necessary to accept the software shall be reviewed and approved. The resulting documentation and associated computer program(s) shall establish the current baseline.
  3. For guidance in dedicating computer programs, see [ASME NQA-1-2015 Part III, Subpart 3.2-2.1](#), *Quality Assurance Requirements for Commercial Grade Items and Services, Commercial Grade Computer Programs, and Software Services* and the references therein.
  4. Subsequent revisions of software from non-NQA- qualified suppliers that do not follow NQA-1 Subpart 2.7 to V&V the software for acceptance (as described in SOFT-V&V), shall be dedicated. See SOFT-GEN for details. Unless a more conservative approach is required by the LANL SRLM, use the following graded approach for dedicating software revisions (changes):
    - a. Dedicate major changes (see Definitions).
    - b. Dedicate minor changes (see Definitions) if the changes affect the critical characteristic(s) that pertain to the functionality of the computer program or the associated SSC.
  5. Incorporate the following guidance as applicable, when dedicating computer programs (Ref. NQA-1-2015, Part III, Subpart 3.2-2.1):
    - a. Acceptable data for historical performance should evaluate the industry-monitored performance of the commercial grade computer program, industry product tests, certification to national codes and standards (nonnuclear-specific), and other industry records or databases. When a computer program has been demonstrated to be

reliable based on its historical performance, it should be credited during dedication. Historical performance should be supported by the use of one of the other verification methods listed above.

- b. This acceptance method [Method 4: Acceptable Supplier Item or Service Performance Record] should have a greater application for the dedication of computer programs used in design or analysis. Computer programs that are commercially available and that have industrywide application may be used successfully hundreds or even hundreds of thousands of times daily. The results of these uses and engineering judgment associated with the acceptance of the computer program should be considered when dedicating the computer program.
- c. Errors reported by the users to the supplier and failures associated with structures, systems, and components may be evaluated as part of the failure analysis investigation. This method is most effective when the supplier provides error reports to the purchaser for applicability and significance evaluation and when the users contact the supplier when computer program errors are suspected. A technical support agreement in the procurement documents provides assurance that there is adequate communication between the supplier and users.

**4.0 DESIGNED SSC SOFTWARE V&V AND SWAU**

**Note:** To minimize text duplication, many of the Non-SSC software subsections that appear later (e.g., 5-7) refer back to this SSC-related subsection with instructions that the SSC-related requirements do not apply. Therefore, to facilitate comprehension, Subsection 4.0 SSC-only requirements are followed by a red “\*” at the end of the text and these should be modified or ignored in accordance with direction in those later Subsections. If additional clarification is needed, contact the Chapter POC.

Table 21.4-3 summarizes the minimum V&V and SWAU activities for designed SSC software. The text that follows it provides additional detail.

Table 21.4-3 Minimum Required V&V Activities - Designed SSC Software					
No	V&V Activity Description	Apply by ML <sup>1</sup>			
		1	2	3	4
1	Design Requirement (Input) Review	R	R	R	G
2	Software Test Plan (SWTP) Development & Review <sup>2</sup>	R	R	G	G
3	Software Design Review <sup>3</sup>	R	R	G	G
4	Software Code Review <sup>4</sup>	R	R	-	-
5	Testing/Cx & Review <sup>*2</sup>	R	R	G	G
6	Software Review and Approval for Use (SWAU)	R	R	G	-

<sup>1</sup> R = Required; G = Required but graded; “-” = Not required.  
<sup>2</sup> See ESM Chapter 15, *Commissioning* for minimum test/commissioning (Cx) requirements based on ML.  
<sup>3</sup> See SOFT-DESIGN for design deliverables.  
<sup>4</sup> Where feasible as determined by the LANL SRLM.

#### 4.1 Design Requirement (Input) Review

- A. How. In accordance with the governing document control process<sup>1</sup>, review and document the review of design (input) requirements to ensure they satisfy the design requirement criteria in SOFT-DESIGN. As guidance for ML-1 through ML-3 software, use the requirement review criteria from [IEEE Std 1012-2012](#), *IEEE Standard for System and Software Verification and Validation*<sup>2</sup>. For CGD of software in accordance with [AP-341-703](#), Commercial Grade Dedication or equivalent, ensure that CGD requirements are addressed.
- B. When. As much as practical, review prior to software design.
- C. Who. The Software Designer (SD), SRLM, LANL Commissioning Authority (LCxA) and FDAR review ML-1 through ML-3 software. \* The FDAR review is not required for ML-4 software.

#### 4.2 Test Plan Development and Review

Tests to verify conformance of a computer program to requirements must be planned and executed. The software test plan (SWTP) may include interim testing to support software design, acceptance testing to support commissioning (Cx) and/or in-use testing to support operations after the software is approved for use (SWAU). Therefore, because the test plan may be broader than acceptance testing, it is not used synonymously with Cx test planning documents. The acceptance testing portion of the test plan however, must be integrated into or with Cx documentation. \*

For post modification testing/post maintenance testing (PMT) of SSCs that do not alter the design requirements of SCCs, see [AP-341-803](#), *Post Modification/Post Maintenance Testing*. \*

- A. How. Plan tests as required by the SSC design, [ESM](#) Chapter 15, *Commissioning (Cx)*<sup>3</sup> and this Subsection. Document in a test plan. \* Review and disposition comments in accordance with the governing document control process.

Address the following in the test plan:

1. Perform at least one acceptance test in a simulated environment (i.e., test bed or bench testing) for ML-1 software. \* For ML-2 software, perform at least one acceptance testing in a simulated environment when specified by the FDAR. \* Perform simulated environment acceptance tests prior to acceptance testing in the actual operating environment. \*
2. As determined by the FDAR for ML-1 and/or ML-2 software, where the simulated environment without hardware cannot adequately test the software, test in an environment with hardware. \*
3. Perform a comprehensive, end-of-development acceptance test in the operating environment<sup>4</sup> prior to use. Include support software tests as appropriate.

<sup>1</sup> See [AP-341-620](#), *Review of LANL Produced Design Documents*; [AP-341-621](#), *Design Authority Technical Review*, [AP-341-622](#), *LANL Review of Designs Produced by External Agencies*, and [AP-321-624](#), *Independent External Design Review*, or approved equivalent.

<sup>2</sup> The V&V tasks/criteria in this standard uses system integrity levels (SILs) for grading instead of MLs; use the following approximate correlation: ML-1 = SIL 4; ML-2 = SIL 3; ML-3 = SIL 2; ML-4 = SIL 1.

<sup>3</sup> See [AP-350-406](#), *Startup and Commissioning* or equivalent for details.

## 4 Section SOFT-V&amp;V: Software Verification and Validation

Rev. 0, 06/23/16

4. As a recommendation and/or as required by other [ESM](#) chapters, plan for tests during various points in the software design (interim tests) in addition to the acceptance test. Interim tests may be informal tests or formal tests to support reviews. Test plan documentation for informal testing is recommended, but not required.
5. Test plan documentation for formal interim tests used to support reviews is required and must satisfy the same applicable criteria as the acceptance test<sup>5</sup>. Include the tests and test results in the acceptance testing documentation.
6. Coordinate and integrate the software test plan and testing with OX and the governing (SSC) work control requirements. \*
7. For ML-1 through ML-3 software, include software testing required after approval for use (in-use tests) that confirms continued acceptable performance in the operating system. Include either directly in the test plan or through reference to other documents. Perform in-use tests after the computer program is installed on a different computer, or when there are significant changes in the operating system.
8. Create a test record form or template that contains, at a minimum, the following:
  - a. SSC(s)/item tested, \*
  - b. date of test,
  - c. tester (name) and/or data recorder,
  - d. type of observation (i.e., type of test),
  - e. results, applicability and acceptability (of the test),
  - f. action taken in connection with any deviations,
  - g. reference to the test methodology/test plan used with associated revision,
  - h. person evaluating and accepting test results,
  - i. a unique test record identifier to allow traceability to the specific item tested (if not subsequently assigned as part of the records management protocols),
  - j. computer program tested including system software, as applicable<sup>6</sup>,
  - k. computer hardware, as applicable
  - l. test equipment and calibrations, as applicable,
  - m. simulation models used, as applicable, and
  - n. test problems/cases.

<sup>4</sup> As determined by the LANL SRLM, where testing in the operating environment introduces unacceptable risk or loss, an exception to performing testing in the operating environment may be pursued in accordance with [ESM](#), Chapter 1, General, Z10, *General Requirements for All Disciplines/Chapters*.

<sup>5</sup> From NQA-1. Note that “same criteria” allows one to employ a subset of the full suite of acceptance test criteria for unit testing (i.e., white box testing); program, subsystem, or system testing (i.e., gray-box testing).

<sup>6</sup> For read-only software, identify computer program name to the extent possible (See SOFT-GEN).

## 4 Section SOFT-V&amp;V: Software Verification and Validation

Rev. 0, 06/23/16

9. Incorporate Appendix A, *Software Test Plan Requirements Table*.
  10. See [IEEE Std 829](#), *IEEE Standard for Software and System Test Documentation* for guidance.
  11. Provide 60%, 90% and 100% test plan submittals for review and approval. See References, *V&V Examples* for an example test plan.
  12. When dedicating software in accordance with [AP-341-703](#), *Commercial Grade Dedication* or equivalent, the acceptance test plan may be a separate test plan or combined with the CGD Technical Evaluation and Acceptance (TEA) Plan. If combined, ensure CGD requirements such as critical characteristics are specified.
  13. Identify and coordinate with associated SSC tests (e.g., qualification tests for seismic qualification of electrical equipment, Ref. [IEEE Std 344](#), *Standard for Seismic Qualification of Equipment for Nuclear Power Generating Stations*.) \*
  14. Review and approve in accordance with the governing document control process. Review for adequacy and conformance to this chapter.
  15. Maintain test plan accuracy through approval for use.
  16. As required, identify hold and/or witness points and associated responsibilities.
  17. For ML-1 and ML-2 software, design tests to demonstrate performance requirements are satisfied and relevant unintended defects are detected. In layman's terms, design tests to try to "break the software". Incorporate the following:
    - a. Incorporate hazards analysis and industry accepted test strategies and techniques.<sup>7</sup>
    - b. Ensure the integrated software/SSC system can handle and recover from errors.
    - c. Demonstrate that loss of operational control (i.e., the computer program crashes) does not occur in normal and abnormal conditions and/or that software errors do not damage or otherwise cause SSC failures or unsafe conditions. \*
- B. When. Develop after software requirements have been established, and concurrent with the software design. Submit draft test plans with draft software design deliverables. Submit final test plan with the final software design and prior to acceptance testing.
- C. Who. A SD develops, reviews and approves; the LCxA reviews; \* the SRLM reviews and approves. Review by a quality assurance subject matter expert (QA SME) is recommended for ML-1 and ML-2.

**Note:** Where possible, the SD should be a subject matter expert (SME) other than the SD that designed the software.

## 4.3 Software Design Review

- A. How. In accordance with the governing document control process<sup>8</sup>, review and document the review to ensure the software design satisfies the software design

<sup>7</sup> See *Certified Software Quality Engineer Handbook*, L. Westfall, American Society of Quality Press.

## 4 Section SOFT-V&amp;V: Software Verification and Validation

Rev. 0, 06/23/16

criteria specified in SOFT-DESIGN. Perform the following as part of the design review:

1. Review for acceptability of the design requirements, the design and conformance to the design process.
2. Evaluate the technical adequacy of the approach;
3. Ensure internal completeness, consistency, clarify, and correctness of the software design.
4. Verify that the software design is compliant with and traceable to the approved software design requirements.
5. Review test results.

As guidance for ML-1 through ML-3 software, use design review criteria from [IEEE Std 1012](#), *IEEE Standard for System and Software Verification and Validation*<sup>9</sup> and [IEEE STD 1028](#), *IEEE Standard for Software Reviews and Audits*.

- B. When. Review in accordance with the software design deliverable submittal schedule and prior to acceptance testing.
- C. Who. The SD and SRLM review. In addition, for ML-1 and ML-2 software, one or more individuals other than those who designed the software must review (Independent Reviewers, or IRs). IRs may be from the same organization.

#### 4.4 Computer Program Code (Code) Review

- A. How. In accordance with the governing document control process, review and document the review to ensure the code satisfies the criteria in SOFT-DESIGN. *Guidance: use the code review criteria from [IEEE Std 1012](#), IEEE Standard for System and Software Verification and Validation.*<sup>10</sup>
- B. When. Review the code prior to acceptance testing.
- C. Who. The Software Designer (SD) and SRLM review. In addition, an independent reviewer (IR) SME other than the SD that developed the code shall review.

#### 4.5 Testing/Cx, Review and Report

- A. How. Test, review, and provide test (also commonly referred to as qualification) reports in accordance with the SSC design, software test plan, and Cx documents. \* These reports are referred to as SWTRs. Ensure the following:
  1. Complete activities to prepare for testing in accordance with the governing (SSC) work control documents and: \*
    - a. Ensure required software backups are accessible in the event that a need to “rollback” to original files arises.
    - b. Ensure that documentation required prior to testing has been received.

<sup>8</sup> See [AP-341-620](#), *Review of LANL Produced Design Documents*; [AP-341-621](#), *Design Authority Technical Review*, [AP-341-622](#), *LANL Review of Designs Produced by External Agencies*, and [AP-321-624](#), *Independent External Design Review* or equivalent.

<sup>9</sup> To apply the V&V tasks/criteria in this standard that uses system integrity levels (SILs) for grading instead of MLs, use the following correlation: ML-1 = SIL 4; ML-2 = SIL 3; ML-3 = SIL 2; ML-4 = SIL 1.

<sup>10</sup> As determined by the LANL SRLM, when it is not feasible for LANL to obtain source code (e.g., for proprietary reasons), review the supplier's code review V&V process and objective evidence.

## 4 Section SOFT-V&amp;V: Software Verification and Validation

Rev. 0, 06/23/16

- c. Review results and resolution of nonconformances identified by prior tests (e.g., Nonconformance reports (NCRs), bug/error reports, etc.). Ensure that issues have been sufficiently resolved to proceed with the test.
- d. Ensure suitable environmental conditions exist throughout the testing process.
- e. As required for integrated software/hardware testing, ensure M&TE such as calipers, thermometers, etc. are in calibration, ready for use, and that proper M&TE maintenance, storage and handling is in place in accordance with [P330-2](#), *Control and Calibration of Measuring and Test Equipment (M&TE)*, or equivalent. \*
- f. Ensure readiness of temporal test requirements (e.g., ensuring witnesses are present and ready when required etc.).
- g. Verify that those performing and/or supporting testing have the minimum training, qualification, authorization, and as appropriate, certifications.
- h. Ensure a clear understanding of the roles and responsibilities for those performing the test(s), as well as those that may be affected by the test(s).
- i. Other activities as may be required by the test plan.
- j. Conduct tests under controlled conditions using established acceptance and performance criteria.
- k. Ensure specified calibration, environmental conditions, and storage requirements are maintained throughout the test process. \*
- l. Ensure configuration requirements are satisfied throughout the duration of the test.
- m. Indicate the status of the test (i.e. tested, not tested, testing in process) indicate on the SSC or in documents traceable to the SSC to ensure that required tests are performed and to ensure SSCs that have not passed tests are not inadvertently used or operated. For ML-1 and ML-2 software, use physical location, tags, markings, shop travelers, stamps, inspection records, or other suitable means to indicate test status. Specify the authority for application and removal of tags, markings, etc.
- n. Retest, as appropriate, modified and/or replaced software.
- o. Ensure the required post-test configuration is in place after completing the test. \*
- p. Document the test results on a test record form to indicate the ability of the software to satisfactorily perform its intended function/performance criteria. Include the minimum test record information on the form or template (see Section 4.2).
- q. As applicable, ensure that CGD requirements are addressed.
- r. Review and evaluate the test results to requirements and ensure requirements have been satisfied. Assure that the software produces correct results. Determine and document the acceptability of the test



## 4 Section SOFT-V&amp;V: Software Verification and Validation

Rev. 0, 06/23/16

in a test (qualification) report (SWTR). Review in accordance with the test plan. Address the following:

- 1) Perform preliminary evaluations to determine the validity of the test results and the appropriateness of further testing. Identify additional test requirements and required changes to the test plan. Retest as required.
  - 2) Analyze and evaluate the data to verify completeness of results, conformance to design (input) requirements, achievement of test objectives, and conformance to the test plan. See References: *V&V Examples* for an example report.
2. Retain test documentation per the governing document control/records management processes. Ensure test records have a unique identifier and are traceable to the software.
- B. When. Conduct tests when specified in the SSC design, test plan, and/or Cx documents and prior to approval for use. \*
- C. Who.
1. For ML-1 – ML-3 software, authorized testers (Ts), as identified in the software plan and/or Cx documents perform tests. \* For ML-4 software, Ts as identified in the SSC design and/or Cx documents perform tests. \*
  2. For ML-1 – ML-3 software, those responsible for review and approval are as indicated in the test plan however must include the following: Lead Tester (T), SD or other SD SME individual familiar with the design detail and intended use of the computer program, SO, LCxA and SRLM. \*
  3. For ML4, authorized Ts shall be facility operations and/or Cx personnel authorized by facility management. The SO, LCxA and SRLM review and approve acceptance tests. \*

#### 4.6 Review and Approval for Use

- A. How. For suppliers, perform software product review and release in accordance with the supplier's quality assurance program. For LANL workers that are approving ML-1 through ML-3 software for use at LANL, review and approve for use (SWAU) in accordance with this Subsection.
1. Prepare a SWAU request package that demonstrates: (a) satisfactory completion of V&V, (b) the applicable operational and maintenance (O&M) controls in SOFT-MAINT are in place, and (c) the software is ready to be used in its intended environment.
  2. Perform a review of the SWAU request package using the criteria from SOFT-V&V and SOFT-MAINT and document the review and SWAU. For ML-1 and ML-2 software, include (either directly or through reference) the intended use and any associated limitations, access controls, etc. for using the software. See SOFT-V&V-FM01, *Software Approval for Use Form (SWAU, Sample) With Example*.
  3. For changes to software in LANL facilities, [AP-341-507](#), *SSC Software Change Package* and [AP-341-801](#), *Post Modification/Post Maintenance Testing* may be used to document SWAU. \*

4 Section SOFT-V&V: Software Verification and Validation

Rev. 0, 06/23/16

- 4. For startup and restart of ML-1 through ML-3 software in LANL nuclear facilities, the operational readiness processes may be used to document SWAU. See [PD115](#), *LANL Readiness Program*.
- B. When. Review and SWAU prior to intended use (operation).
- C. Who. The SO prepares the SWAU request package. The SRLM reviews and approves ML-1 through ML-3 software for use. If [AP-341-507](#), [PD115](#), or other processes are used for approval for use, the reviews and approvers are as specified in the respective process. \*

5.0 READ-ONLY SSC SOFTWARE V&V AND SWAU

This Subsection applies to LANL workers that acquire Read-Only SSC software. Read-Only software is software where the computer program cannot be changed other than through replacement of the computer program and/or the associated SSC<sup>11</sup>.

Table 21.4-4 presents the minimum V&V and SWAU activities that must be performed for Read-Only SSC software. The following text provides additional detail.

Table 21.4-4 Minimum Required V&V Activities - Read Only SSC Software					
No	V&V Activity Description	Apply by ML <sup>1</sup>			
		1	2	3	4
1	Requirement (Input) Review	R	R	G	G
2	Test Plan Development and Review <sup>2</sup>	R	R	G	G
3	Design Review	R	R	G	G
4	Testing/Cx and Review <sup>2</sup>	R	R	G	G
5	Review and Approval for Use (SWAU)	R	R	G	-

<sup>1</sup> R = Required; G = Required but graded; - = Not required.  
<sup>2</sup> See [ESM](#) Chapter 15, *Commissioning* for minimum test/commissioning (Cx) requirements based on ML.

5.1 Requirement (Input) Review

Read Only SSC software input requirements are specified in the SSC design and/or acquisition requirement (input) documents. Examples include a Requirements Criteria Document (RCD) and Statement of Work (SOW). Prior to acquisition, review the SSC requirement to the requirement criteria of the respective document in accordance with the governing document control process. As applicable, ensure CGD requirements are addressed.

5.2 Test Plan Development and Review

Use the SSC Cx documents and SSC design as the software test plan documents. Apply the General Test Plan Requirements in Table 21.4-A1 to Cx as shown in the table. Use the remaining Subsection 4.2 requirements as Cx guidance. Review and approve the Cx documents per [ESM Chapter 15](#) prior to testing.

<sup>11</sup> Facility/system control software is considered designed SSC software for the purposes of this chapter.

4 Section SOFT-V&V: Software Verification and Validation

Rev. 0, 06/23/16

5.3 Design Review

Review the Read-Only software as part of the review of the associated SSC design in accordance with the SSC design deliverable schedule in ESM [Chapter 1](#), Z10. Perform the review in accordance with the governing SSC design review process.<sup>12</sup>

For ML-1 and ML-2 software, specify and review the supplier’s V&V documentation of the software product. Review based on Subsection 4 criteria and as applicable, commercial grade dedication (CGD) requirements.

5.4 Testing/Cx and Review

Test per the SSC Cx and SSC design documents. Provide, review and approve Cx report per ESM [Chapter 15](#) prior to approval for use.

5.5 Review and Approval for Use

See Subsection 4.6 and apply to Read-Only SSC software.

6.0 DESIGNED NON-SSC SOFTWARE V&V AND SWAU

Table 21.4-5 summarizes the minimum V&V and SWAU activities for designed Non-SSC software.

Table 21.4-5 Minimum Required V&V Activities –Designed Non-SSC Software					
No	V&V Activity Description	Apply by ML <sup>1</sup>			
		1	2	3	4
1	Model V&V <sup>2</sup>	R	R	G	G
2	Design Requirement Review	R	R	R	R
3	Test Plan (SWTP) Development and Review	R	R	G	-
4	Design Review <sup>2</sup>	R	R	G	G
5	Code Review <sup>4</sup>	R	R	-	-
6	Testing & Review	R	R	G	G
7	Review and Approval for Use (SWAU)	R	R	G	-

<sup>1</sup> R = Required; G = Required but graded; “-” = Not required.  
<sup>2</sup> Required for software that includes models.  
<sup>3</sup> See SOFT-DESIGN for design deliverables.  
<sup>4</sup> Where feasible as determined by the LANL SRLM.

The following text provides additional detail.

<sup>12</sup> See [AP-341-620](#), Review of LANL Produced Design Documents; [AP-341-621](#), Design Authority Technical Review, and [AP-341-622](#), LANL Review of Designs Produced by External Agencies.

## 4 Section SOFT-V&amp;V: Software Verification and Validation

Rev. 0, 06/23/16

## 6.1 Model V&amp;V

- A. How. For software that includes models (see SOFT-DESIGN), verify and validate the appropriateness of the model in addition to the appropriateness of the computer code used to provide the outcome of the model. In addition to the requirements of this subsection, use industry accepted model V&V methods.<sup>13,14</sup> Example methods include alternate calculations, references to previously established standards and practices, and specialized peer reviews. Document the model V&V in addition to the computer code V&V as part of the software design and testing reviews.
- B. When. After requirements and concurrent with design.
- C. Who. The SRLM reviews and approves.

## 6.2 Design Requirement (Input) Review

See Subsection 4.1 and apply the following clarifications:

- Requirements associated with SSCs do not apply.
- FDAR and LCxA review are not required.

## 6.3 Test Plan Development and Review

See Subsection 4.2 and apply the following clarifications:

- Requirements associated with SSCs, including commissioning (Cx) and governing SSC work controls do not apply.
- Acceptance testing in a simulated environment (i.e., test bed) does not apply.

## 6.4 Software Design Review

See Subsection 4.3.

## 6.5 Designed Non-SSC Computer Program Code Review

See Subsection 4.4.

## 6.6 Testing and Review

See Subsection 4.5 and apply the following clarification:

- Requirements associated with SSCs, including testing in simulated environment, commissioning (Cx), governing SSC work controls and M&TE do not apply.

## 6.7 Review and Approval for Use

See Subsection 4.6 and apply the following clarifications:

- Requirements associated with SSCs including [AP-341-507](#), do not apply.
- For changes of ML-1 through ML-3 software in LANL facilities, use SOFT-GEN-FM02, *Non-SSC Software Change Package Form (SWNCP)* or equivalent.

<sup>13</sup> LA-14167-MS, Los Alamos National Laboratory, *Concepts of Model Verification and Validation*, Thacker et al., 2004.

<sup>14</sup> ANSI/ANS 10.7-2013, *Non-Real-Time, High-Integrity Software for the Nuclear Industry – Developer Requirements*.

## 4 Section SOFT-V&amp;V: Software Verification and Validation

Rev. 0, 06/23/16

## 7.0 ACQUIRED NON-SSC SOFTWARE V&amp;V AND SWAU

Table 21.4-5 presents the minimum V&V and SWAU activities that must be performed for acquired Non-SSC software. The following text provides additional detail.

Table 21.4-5 Minimum Required V&V Activities – Acquired Non-SSC Software					
No	V&V Activity Description	Apply by ML <sup>1</sup>			
		1	2	3	4
1	Model V&V <sup>2</sup>	R	R	G	G
2	Acquisition Requirement (Input) Review	R	R		R
3	Test Plan (SWTP) Development and Review	R	R	G	-
4	Acquisition Review	R	R	G	G
5	Testing & Review	R	R	G	G
6	Review and Approval for Use (SWAU)	R	R	G	-

<sup>1</sup> R = Required; G = Required but graded; \* = Not Required.  
<sup>2</sup> Required for software that includes models.

## 7.1 Model V&amp;V

See Subsection 6.1.

## 7.2 Acquisition Requirement (Input) Review

A. How.

- In accordance with the governing acquisition and associated document control processes<sup>15</sup>, review the acquisition documents (e.g., Statement of Work, Exhibit H etc.) to ensure they satisfy the acquisition criteria in the SOFT-ACQUIRE. As applicable, ensure CGD requirements are addressed.

B. When. Review prior to software acquisition.C. Who. The SO and SRLM review.

## 7.3 Test Plan Development and Review

See Subsection 4.2 and apply the following clarifications:

- Requirements associated with SSCs, including commissioning (Cx) and governing SSC work controls do not apply
- Acceptance testing in a simulated environment (i.e., test bed) does not apply
- Testing associated with software design (e.g., interim tests) do not apply
- Preparation or incorporation of a hazard analysis is not required
- The test plan is prepared, reviewed and approved in accordance with the governing document control process

Supplier-provided test cases, documentation and comparative alternate calculations/published data may be used as appropriate to demonstrate the software meets requirements for its intended use.

<sup>15</sup> See P840-1, *Quality Assurance for Procurements* or equivalent.

## 4 Section SOFT-V&amp;V: Software Verification and Validation

Rev. 0, 06/23/16

## 7.4 Acquisition Review

- A. How. For ML-1 through ML-3 software, review the acquired software for conformance to the acquisition documents (see SOFT-ACQUIRE). Evaluate the technical adequacy and completeness and verify that the software is traceable to the acquisition requirements. As applicable, review in accordance with CGD requirements. As applicable, ensure valid software licenses and markings that indicate original manufacturer's product to help prevent acquisition of suspect/counterfeit software.
- B. When. Review prior to acceptance testing.
- C. Who. The SO performs the acquisition review.

## 7.5 Testing and Review

See Subsection 6.6 with the clarification that requirements associated with software design (e.g., interim testing) do not apply.

## 7.6 Review and Approval for Use

See Subsection 6.7.

## 8.0 TOOLBOX CODE V&amp;V AND SWAU

- A. How. V&V toolbox codes (See SOFT-ACQUIRE for definition and information) in accordance with the instructions for the specific toolbox code. Review and SWAU per Subsection 6.7.
- B. When. Complete for each installation of a toolbox code.
- C. Who. The SO performs and the SO and SRLM review and approve.

## 9.0 SIMPLE &amp; EASILY UNDERSTOOD NON-SSC SOFTWARE V&amp;V/SWAU

- A. How. Perform V&V/SWAU on "simple and easily understood" Non-SSC software used in the design of the SSCs per Subsections 6 and 7 (pre-verify), or V&V/SWAU as part of the SSC design. V&V for each time it is used (individually verify).
  - 1. To individually verify:
    - a. Follow the SSC design process. Document the design analysis (e.g., [EN 341-605](#), *Calculations*, for LANL). Include the following:
      - 1) the objective of the analyses
      - 2) design inputs and their sources
      - 3) results of literature searches or other applicable background data
      - 4) assumptions and indication of those assumptions that must be verified as the design proceeds.
      - 5) identification of any computer calculation, including identification of the computer type, computer program name, and revision, inputs, outputs, evidence or reference to computer program verification, and the bases (or reference thereto) supporting application of the computer program to the specific physical problem.
      - 6) review and approval.

## 4 Section SOFT-V&amp;V: Software Verification and Validation

Rev. 0, 06/23/16

- b. The following software information should be included in the SSC design verification documentation and retained as a record: (e.g., calculation documentation): (a) design inputs, (b) the computer-program-generated results, (c) computer-generated evidence of the programmed algorithms or equations (e.g., computer program listings, spreadsheet cell contents).
  - c. Those performing individual software verifications must be cognizant that software may not have undergone CGD and/or may not have been purchased from an NQA-1 supplier. Accordingly, they must apply the appropriate rigor in the SSC design review to ensure the computer program produces satisfactory results. Show that the software produces correct solutions for the encoded mathematical model within defined limits for each parameter employed. As applicable, show that the encoded mathematical model produces a valid solution to the physical problem associated with the particular application.
2. To SWAU, see Subsection 6.7.
- B. When. V&V/SWAU the software each time the software is used for a SSC design, when required by the SSC design V&V process, and prior approval of the associated SSC design.
  - C. Who. As per the applicable SSC design V&V process.

**APPENDICES, FORMS AND REFERENCES**

## APPENDICES:

APPENDIX A: SOFTWARE TEST PLAN REQUIREMENTS TABLE

## FORMS:

SOFT-V&amp;V-FM01: SOFTWARE APPROVAL FOR USE FORM (SWAU)

## REFERENCES: V&amp;V EXAMPLES

The reference collection associated with SOFT-V&V contains example V&V documentation. Examples initially provided are for informational purposes only and have not been reviewed for compliance to the requirements of this chapter. Better examples will be posted as they become available. At time of writing, they included:

- R.1 Example Requirements Traceability Matrix
- R.2 Example No. 1 Test Plan
- R.3 Example No. 1 Test Report
- R.4 Example No. 2 Test Report

**APPENDIX A: COMPUTER PROGRAM TEST PLAN REQUIREMENTS**

Table 21.4-A1 provides computer program test plan requirements. The text in *italicized print* should be used as guidance for all items marked as “G”.

**Table 21.4-A1 Computer Program Test Plan Requirements**

No	Requirement	Apply by ML <sup>1</sup>			
		1	2	3	4
<b>1.0 General Test Plan Requirements<sup>6</sup></b>					
1.01	Include or reference the test name, objective(s) and date.	R	R	R	*
1.02	Include or reference the test configuration ( <i>including pre-test, during testing, and post-test configurations where appropriate</i> ).	R	R	G <sup>2</sup>	*
1.03	Satisfy test prerequisites. Prerequisites include the following as applicable: calibrated instrumentation, appropriate and/or special equipment, trained personnel, condition of test equipment and the item to be tested, suitable environmental conditions, and provisions for data acquisition. Ensure suitability and readiness of equipment and/or algorithms used for simulated computer program test inputs/fault seeding.	R	R	R	*
1.04	Assure that suitable environmental conditions are satisfied during testing	R	R	R	*
1.05	Assure adequate instrumentation is available and used.	R	R	R	*
1.06	Assure appropriate tests and equipment are used (e.g., performance of a pneumatic leak test may be required because a hydrostatic leak test could damage the item). Select measuring and test equipment based on the type, range, accuracy, and tolerance needed to accomplish the required measurements for determining conformance to specified requirements.	R	R	R	*
1.07	Provide for traceability of measuring and test equipment to the test.	R	R	R	*
1.08	Assure necessary monitoring is performed.	R	R	R	*
1.09	Specify characteristics to be tested.	R	R	R	*
1.10	Obtain test requirements and acceptance criteria <i>from, or approved by, the responsible design organization</i> .	R	R	G	*
1.11	Control the test. (This includes controlling changes during testing; includes as appropriate, software design verification, factory acceptance, site acceptance, and in-use tests.)	R	R	R	*
1.12	Control tests under appropriate environmental conditions using the tools and equipment necessary to conduct the test in a manner to fulfill test requirements and acceptance criteria.	R	R	R	*
1.13	Develop/specify test methods to obtain necessary data with sufficient accuracy for evaluation and acceptance.	R	R	R	*
1.14	Base test requirements and acceptance criteria upon specified requirements contained in the applicable design documents, or other pertinent technical documents that provide approved requirements. Ensure acceptance criteria are specific, unambiguous, measurable, and attainable.	R	R	R	*
1.15	Document and maintain test results.	R	R	R	*
1.16	When changes to the approved configuration of a facility are required for testing purposes, obtain approval by the Facility Design Authority (DA) or FDAR, and others as required in the governing work control process (e.g., <a href="#">AP-341-504</a> , <i>Temporary Modification Control</i> ), prior to performing the test.	R	R	R	*
1.17	When identification and traceability requirements are specified in requirements documents, design the test to satisfy the requirements.	R	R	R	*



## 4 Section SOFT-V&amp;V: Software Verification and Validation

Rev. 0, 06/23/16

No	Requirement	Apply by ML <sup>1</sup>			
		1	2	3	4
1.18	When post-installation testing is planned, mutually establish post-installation test requirements and acceptance documentation with the item purchaser and supplier.	R	R	R	*
1.19	When special process examinations/tests are performed (e.g., Nondestructive Examination (NDE) such as radiographic testing, magnetic particle testing, ultrasonic testing, liquid penetrant testing, electromagnetic testing, neutron radiographic testing, leak testing, acoustic emission testing, and visual testing), address the applicable requirements of <a href="#">STD-342-100, Engineering Standards Manual</a> , Chapter 13, <i>Welding, Joining &amp; NDE</i> , and <a href="#">P330-5, Special Processes</a> .	R	R	R	*
1.20	Perform testing in accordance with the governing work control process and the test plan; address applicable health, safety, environmental and security requirements associated with the testing. Ensure post-configuration conditions meet post-configuration test plan requirements, including approvals as required.	R	R	R	*
1.21	Evaluate conformance of test results with test requirements and acceptance criteria.	R	R	R	*
1.22	Identify those responsible for accepting the test(s).	R	R	R	*
1.23	The process for obtaining a consent to waive specified hold points (testing, inspection and/or witness holds) or, make changes to the test plan, must be as described in applicable work control documents (e.g., field change, design change process, etc.) that govern configuration management for the item. If the waiver and/or change process is not addressed in the governing work control process, then address the waiver process, including obtaining consent from authorized personnel, in the test plan.	R	R	R	*
1.24	Obtain consent to waive hold points prior to continuation of work beyond the designated hold point and ensure documentation of the consent.	R	R	R	*
1.25	As applicable, develop and prepare an approach to sampling. <i>When used, base sampling on standard statistical methods with approval from the responsible engineer for the item. Obtain support from the responsible engineer and/or quality assurance representative in development of the plan as required. When used for accepting commercial grade items and services, base sampling upon standard statistical methods with supporting engineering justification. Consider lot/batch/traceability, homogeneity, and the complexity of the item.</i> If sampling is performed, document the sampling method and the approval of the method by the responsible engineer. When less than 100% sampling is used, draw samples at random.	R	R	G	*
1.26	<i>Demonstrate adequacy of performance under conditions that simulate the most adverse design conditions with operating modes and environmental conditions considered in determining the most adverse conditions.<sup>4</sup></i>	R	R	G	*
1.27	<i>Where the tests intended to verify only specific design features, the other features of the design shall be verified by other means. Establish and verify scaling laws when tests are being performed on models or mockups.<sup>4</sup></i>	R	G	G	*
1.28	<i>Perform error analysis, where applicable, on the results of model test work prior to the use of the final design.<sup>4</sup></i>	R	G	G	*
<b>2.0 Additional Test Plan Requirements for General Computer Program Testing<sup>6</sup></b>					
2.01	<i>Involve engineering, in accordance with the governing work control process, as applicable, in safety software inspection, including obtaining concurrence to the acceptability of safety software verification results.</i>	R	R	G	*
2.02	<i>Obtain test requirements and acceptance criteria by the organization responsible for the use of the computer program.</i>	R	R	G	*

## 4 Section SOFT-V&amp;V: Software Verification and Validation

Rev. 0, 06/23/16

No	Requirement	Apply by ML <sup>1</sup>			
		1	2	3	4
2.03	Include required tests, test sequence and frequency.	R	R	R	*
2.04	Include required ranges of input parameters.	R	R	R	*
2.05	Identify the stages at which testing is required.	R	R	R	*
2.06	<i>Include criteria for establishing test cases.</i>	R	R	G	*
2.07	<i>Include requirements for testing logic branches.</i>	R	R	R	*
2.08	Include requirements for hardware integration.	R	R	R	*
2.09	Include anticipated output values.	R	R	R	*
2.10	Include acceptance criteria, reports, records, standard formatting, and conventions.	R	R	R	*
2.11	Demonstrate the adherence of the computer program to documented requirements.	R	R	R	*
2.12	<i>Evaluate technical adequacy through comparison of test results from alternative methods such as hand calculations, calculations using comparable proven programs, or empirical data and information from technical literature.<sup>3</sup></i>	R	R	G	*
2.13	<i>For software design verification testing, demonstrate the capability of the computer program(s) to provide valid results for test problems encompassing a range of documented permitted usage.<sup>4</sup></i>	R	R	G	*
2.14	For those computer programs used for operational control, demonstrate required performance over the range of operation of the controlled function or process.	R	R	R	*
2.15	<i>For computer programs in which computer program errors, data errors, computer hardware failures, or instrument drift can affect required performance, prescribe and perform periodic in-use manual or automatic self-check tests.<sup>5</sup></i>	R	R	G	*
2.16	Provide for system software testing. <i>For new or revised software, evaluate, review, test, and accept for use. Place system software under configuration change control. Evaluate system software changes for impact on the software product to determine the required level of retesting.</i>	R	R	G	*
<b>3.0 Additional Test Plan Requirements for Computer Program Acceptance Testing<sup>7</sup></b>					
3.01	Include the process of exercising or evaluating a system or system component by manual or automated means to ensure that it satisfies the specified requirements and to identify differences between expected and actual results in the operating environment.	R	R	R	*
3.02	Demonstrate that the computer program adequately and correctly performs all intended functions (i.e., specified software design requirements).	R	R	R	*
3.03	Demonstrate that the computer program properly handles abnormal conditions and events as well as credible failures.	R	R	G	*
3.04	Demonstrate that the computer program does not perform adverse unintended functions.	R	R	R	*
3.05	<i>Demonstrate that the computer program does not degrade the system either by itself or in combination with other functions or configuration items.</i>	R	R	G	*
3.06	Ensure that acceptance testing is performed and passed prior to approval of the computer program for use.	R	R	R	*
3.07	Ensure that configuration items are identified and are under configuration change control prior to starting acceptance testing.	R	R	R	*
3.08	Ensure that acceptance testing is planned, performed, and passed for all software design requirements.	R	R	G	*

4 Section SOFT-V&V: Software Verification and Validation

Rev. 0, 06/23/16

No	Requirement	Apply by ML <sup>1</sup>			
		1	2	3	4
3.09	Ensure that acceptance testing ranges from a single test of all software design requirements to a series of tests performed during computer program development. (Performance of a series of tests provides assurance of correct translation between activities and proper function of individual modules).	R	R	R	*
3.10	Include a comprehensive acceptance test that is performed in the operating environment prior to use.	R	R	R	*
3.11	<i>Provide for documentation and disposition of observations of unexpected or unintended results prior to test approval.</i>	R	R	G	*
3.12	For acceptance testing of changes to a computer program, provide for selective retesting to detect unintended adverse effects introduced during changes to the computer program. Provide assurance that the changes have not caused unintended adverse effects in the computer program, and to verify that a modified system (s) or system component (s) still meets specified software design requirements.	R	R	R	*

Notes:

<sup>1</sup> R = Required; G = Graded by using *italicized text* as guidance and non-italicized text as required; \* = As required by [ESM](#) Chapter 15, *Commissioning* and the SRLM.

<sup>2</sup> Not used.

<sup>3</sup> Applies to Non-SSC software only.

<sup>4</sup> Applies to design verification testing only, reference NQA-1, Part I, Requirement 3, *Design Control*.

<sup>5</sup> Apply as required during testing to ensure that test results are valid.

<sup>6</sup> From NQA-1, Part I, Requirement 11, *Test Control*. As applicable, General Test Plan Requirements apply to testing of items as graded (items that include computer programs or do not include computer programs). Additional Test Plan Requirements for General Computer Program Testing apply to formal interim computer program tests and computer program acceptance tests as graded.

<sup>7</sup> From NQA-1, Part II, Subpart 2.7, *Quality Assurance Requirements for Computer Software for Nuclear Facility Applications* and associated guidance. As applicable, additional Test Plan Requirements for Computer Program Acceptance Testing apply to acceptance testing of computer programs in addition to requirements under Headings 1 and 2 in this table as graded in the table.

This search tool may not include the latest files; use individual files for work.



# Conduct of Engineering Software Approval for Use Form (SWAU)

SWAU No.:

Rev.:

1.0 GENERAL INFORMATION					
1.1 Software Name and Revision:					
1.2 TA Number(s):		1.3 Facility Number(s):		1.4 Facility Name(s):	
1.5 SWID No.:			1.6 Equipment ID:		
1.7 Software Intended Use:					
1.8 SSC Software/Non-SSC Software:		SSC <input type="checkbox"/> Non-SSC <input type="checkbox"/>			
1.9 Highest ML Associated With the Software		ML-1 <input type="checkbox"/> ML-2 <input type="checkbox"/> ML-3 <input type="checkbox"/> ML-4 <input type="checkbox"/>			
1.10 Software Type		Safety Software <input type="checkbox"/> Non-Safety Software <input type="checkbox"/>			
2.0 APPROVALS					
2.1 As the Software Responsible Line Manager (SRLM), I attest that the software approval for use requirements of ESM Chapter 21, <i>Software</i> have been satisfied and that the software is ready to be used in its intended operating environment in accordance with approved operating and maintenance instructions and, any limitations as described in Section 3.					
Printed Name/Z No.:		Signature			Date
3.0 LIMITATIONS					
3.1 Limitations					
4.0 REVISIONS					
4.1 Revision	4.2 Date	4.3 Change Description and Reason for Change			
5.0 ATTACHMENTS					
5.1 Attachment No.	5.2 Attachment Date	5.3 Attachment Title			

This search tool may not include the latest files. Use individual files for work.

## LANL



# Conduct of Engineering Software Approval for Use Form (SWAU) Instructions with Example

## GENERAL

These instructions are for completing the form and are not part of the completed form.

LANL personnel: Endeavor to use Chapter forms as-is and report issues and improvement ideas to the Chapter 21 POC. POC may authorize other methods equivalent to chapter forms in writing.

LANL subcontractors must use Chapter forms to satisfy Chapter requirements for SSC software. For Non-SSC software, subs may either use their own forms or integrate, adapt, and reformat the forms; either approach is acceptable so long as key functions, data, and approvals are retained.

## HEADER

Field	Entry Information
Software Name	Enter the name of the software in the upper right hand corner of the header.
SWAU No.	The numbering scheme of this form follows that of the SRM's document control/records management system. For software used in the operation of existing or new LANL facilities, use numbering scheme and process specified in <a href="#">AP-341-402</a> , <i>Engineering Document Management in Operating Facilities</i> .
Rev.	Enter the SWAU revision number. Initial issue is 0.

## 1.0 GENERAL INFORMATION

Field	Entry Information
1.1	Enter the name and revision number of the software.
1.2	Enter the technical area(s) where the software is used. If the software is used site wide enter "site wide".
1.3	Enter the facility number(s) where the software is used. If the software is used site wide enter "site wide".
1.4	Enter the facility (building) name(s) where the software is used. If used site wide enter "site wide".
1.5	Enter the software identification number (SWID). The SWID should be the same SWID as on the Form <a href="#">2033</a> . SWIDs are obtained in accordance with <a href="#">AP-341-402</a> , <i>Engineering Document Management in Operating Facilities</i> .
1.6	For SSC software, enter the equipment identification number associated with the software per ESM Chapter 1, Section 200 (e.g., 03-0216-HVAC-BAS-1). For Non-SSC software, enter NA.
1.7	Provide why the software is needed (justification) and describe what the software does.
1.8	Indicate whether the software is SSC Software or non-SSC software by checking the appropriate box. Check only one box.
1.9	Indicate the highest management level (ML) associated with the software. Reference <a href="#">AP-341-502</a> . (ML-1 is the highest ML indication). If the software associated with multiple MLs, check the highest ML box. Check only one box.

## LANL

Field	Entry Information
1.10	Indicate whether the software is safety software or non-safety software by checking the appropriate box. Reference Form <a href="#">2033</a> . If the software is used for both, check the Safety software box. Check only one box.

## 2.0 REVIEW AND APPROVAL

Field	Entry Information
2.1	<p>The SRLM reviews the SWAU request package (See SOFT-V&amp;V) to ensure, satisfactory completion of V&amp;V, applicable operational and maintenance (O&amp;M) controls per SOFT-MAINT are in place, and the software is ready to be used in its intended environment.</p> <p>The SRLM attests that: (a) the SWAU requirements of ESM Chapter 21, <i>Software</i> have been satisfied; and (b) the software is ready to be used in its intended operating environment in accordance with approved operating and maintenance instructions and any limitations as described in Section 3. Enter the SRLM name, Z number if applicable, organization, signature, and date.</p> <p><b>Note:</b> For assistance in determining the SRLM, see SOFT-GEN Appendix B: <i>SO and SRLM Decision Diagram for FAC-COE</i>.</p>

## 3.0 LIMITATIONS

Field	Entry Information
3.1	Enter any limitations associated with using the software either directly and/or through reference to other documents that define the limitations.

## 4.0 REVISIONS

Field	Entry Information
4.1	If the SWAU requires revision, then revise in accordance with the governing document control program. Enter the revision number, 1, 2, etc. as appropriate.
4.2	Enter the revision approval date.
4.3	Describe the change and why the change was made.

## 5.0 ATTACHMENTS

Field	Entry Information
5.1	Attach the SWAU request package and SRLM request package review documentation. Enter the attachment number(s). List and include other attachments as appropriate.
5.2	Enter the attachment date.
5.3	Enter the attachment title, including revision as appropriate.

## LANL

TABLE OF CONTENTS

1.0 PURPOSE, SCOPE AND APPLICABILITY .....1  
 2.0 KEY DEFINITIONS AND ACRONYMS .....1  
 3.0 SSC SOFTWARE INVENTORY .....1  
 4.0 NON-SSC SOFTWARE INVENTORY .....4

Rev	Date	Description	POC	RM
0	06/23/16	Initial issue	Tobin Oruch, ES-DO	Mel Burnett, CENG-OFF

1.0 PURPOSE, SCOPE AND APPLICABILITY

This section describes the “how, when, and who” for inventorying software. See Table 21.5-1 for a summary of this section. See Chapter 21 applicability in SOFT-GEN.

2.0 KEY DEFINITIONS AND ACRONYMS

See SOFT-GEN for key definitions and acronyms.

3.0 SSC SOFTWARE INVENTORY

- A. How. Inventory **safety** SSC software in the LANL Institutional-level safety software inventory in accordance with [QPA-DO-FSD-001](#), *LANL Safety Software Inventory Procedure*. **Non-safety** SSC software is not required to be inventoried.

**Note:** See ESM Chapter 21, SOFT-GEN and Form [2033](#) to determine if the software is safety software.

- 1. Provide required information, namely:
  - a. Software name,
  - b. Description,  
Version identifier,
  - d. Category,
  - e. Software Risk Level (SRL),
  - f. Application,
  - g. Software Responsible Line Manager (SRLM), and
  - h. Other information as required by [QPA-DO-FSD-001](#).

This search tool may not include the latest files; use individual files for work.

5 Section SOFT-INV: Inventory

Rev. 0, 06/23/16

Table 21.5-1 SOFT-INV Section Summary								
(This table is a summary only and does not include all requirement details. See text for details.)								
Activity No.	SQM Activity	Software Type <sup>1</sup>		Implementation Detail			P1040 R9 Ref. Section	ESM Ch. 21 Ref. <sup>5</sup>
		Safety	Non-Safety	How	When <sup>2</sup>	Who <sup>3, 4</sup>		
<b>1. For SSC Software</b>								
1	Inventory <b>safety</b> software	R	-	<ul style="list-style-type: none"> <li>Provide safety software inventory information</li> <li>Inventory safety software Per <a href="#">QPA-DO-FSD-001</a>, <i>LANL Safety Software Inventory Procedure</i></li> </ul>	<ul style="list-style-type: none"> <li>Provide SSC software inventory information per ESM Chapter 1, Z10 submittal schedule</li> <li>Inventory as soon as practical after approval for use</li> <li>Revise as needed to ensure the inventory is complete and accurate</li> </ul>	<ul style="list-style-type: none"> <li>Design Agency provides safety SSC software inventory information in PEL for designed and acquired software; see SOFT-GEN for Subcontractor responsibilities (D)</li> <li>SRLM inventories safety software (D)</li> </ul>	3.3.7	SOFT-GEN, 3  SOFT-INV, 3
<b>1. For Non-SSC Software</b>								
1	Inventory <b>safety and non-safety</b> software	R	R	<ul style="list-style-type: none"> <li>Inventory safety software in the Institutional-level safety software inventory per <a href="#">QPA-DO-FSD-001</a>, <i>LANL Safety Software Inventory Procedure</i></li> <li>ES-Div: Inventory safety and non-safety software to the Non-SSC software inventory as required by task instruction (future); provide Form <a href="#">2033</a> to ESSC with request to inventory</li> </ul>	<ul style="list-style-type: none"> <li>Inventory as soon as practical after approval for use</li> <li>Revise as needed to ensure the inventory is complete and accurate</li> </ul>	<ul style="list-style-type: none"> <li>Design Agency provides safety SSC software inventory information in PEL for designed and acquired software; see SOFT-GEN for Subcontractor responsibilities (D)</li> <li>SRLM inventories software (D)</li> <li>ESSC supports ES Non-SSC software inventory additions/changes (S)</li> </ul>	3.3.7	SOFT-GEN, 3  SOFT-INV, 4

This search tool may not include the latest files. Use individual files for work.



5 Section SOFT-INV: Inventory

Rev. 0, 06/23/16

Table 21.5-1 SOFT-INV Section Summary (This table is a summary only and does not include all requirement details. See text for details.)								
Activity No.	SQM Activity	Software Type <sup>1</sup>		Implementation Detail			P1040 R9 Ref. Section	ESM Ch. 21 Ref. <sup>5</sup>
		Safety	Non-Safety	How	When <sup>2</sup>	Who <sup>3, 4</sup>		
Notes: <sup>1</sup> See SOFT-GEN and Form <a href="#">2033</a> to determine if software is safety software or non-safety software. R = Required, "-" = Not required. <sup>2</sup> "When" specifies when the activity must be performed. <sup>3</sup> D = Develop or execute. S = Support. <sup>4</sup> SRLM = Software Responsible Line Manager. ESSC = Engineering Services (ES) Software Coordinator as indicated on the <a href="#">ES Division Office (ES-DO) website</a> . See SOFT-GEN (e.g., flow diagram) for assistance in identifying the SRLM. <sup>5</sup> Ref. = ESM Chapter 21 section name and subsection number (e.g., Section SOFT-GEN, Subsection 3, Software Identification and Determination).								

This search tool may not include the latest files; use individual files for work.

## 5 Section SOFT-INV: Inventory

Rev. 0, 06/23/16

**Note:** Do not enter classified information into the inventory.

2. Provide changes to the inventory to ensure completeness and accuracy.
  3. Subcontractors that provide SSC software to LANL must provide minimum required SSC Safety Software inventory information as part of the Project Equipment List (PEL) or via other submittal per ESM Chapter 1, Section Z10.
  4. If LANL has access to and/or configuration control of SSC safety software other than through replacement of the SSC<sup>1</sup>, then it must be inventoried on the Institutional-level safety software inventory.
  5. SSC safety software may be inventoried as a software collection. For example, there may be various software elements – control system executable, Human Machine Interface (HMI) software, field element embedded software – associated with a gas handling process control system software. The software may be inventoried as one item, the gas handling process control system software. Note that the various software elements must be baselined and under configuration management per SOFT-GEN.
- B. When. Inventory SSC safety software as soon as practical after it is approved for use and as needed to ensure the inventory is complete and accurate. Provide inventory information in accordance with the PEL or other submittals schedule per ESM Chapter 1, Section Z10.
- C. Who. Design Agency provides safety SSC software inventory information in PEL for designed and acquired software; see SOFT-GEN for Subcontractor responsibilities. The SRLM<sup>2</sup> is responsible for inventorying safety SSC software.

#### 4.0 NON-SSC SOFTWARE INVENTORY

- A. How.
1. Inventory **safety** Non-SSC software in the Institutional-level safety software inventory in accordance with [QPA-DO-FSD-001](#).
    - a. Subcontractors: Safety Non-SSC software used by a subcontractor and prescribed or otherwise controlled by LANL or DOE (e.g., toolbox codes) must be inventoried. The Subcontractor shall provide the minimum required safety software information to inventory such software in the institutional-level safety software inventory.
  2. ES-Div: Follow the ES-Div inventory process (e.g., future desk instruction), e.g., inventory **safety and non-safety** Non-SSC software in the Engineering Services (ES) Non-SSC software inventory as specified by the ES Software Coordinator (ESSC). *Guidance: As applicable, the SRLM should indicate whether the software is simple and easily understood software used in SSC designs (see definitions). The SRLM should provide the Form 2033 to the ESSC if not previously provided. See the [ES Division website](#) for ESSC*

<sup>1</sup> Includes LANL service contracts where suppliers may upgrade/flash new software on SSCs.

<sup>2</sup> The activity may be delegated however the responsibility is retained by the SRLM.

## 5 Section SOFT-INV: Inventory

Rev. 0, 06/23/16

contact information (e.g., [ES-Software@lanl.gov](mailto:ES-Software@lanl.gov)) and the ES Non-SSC software inventory.

**Note:** As described in the SOFT-MAINT, the software should not be used unless it is listed on the ES inventory or otherwise approved for use (SWAU). The listing of the software on the ES inventory is the primary method to communicate SWAU.

- B. When. Inventory Non-SSC software (both safety and [for ES-Div] non-safety) concurrent with SWAU and as needed thereafter to ensure completeness and accuracy.
- C. Who. The SRLM is responsible for inventorying Non-SSC safety software (for ES-DIV, with support from the ESSC).

*This search tool may not include the latest files; use individual files for work.*

**TABLE OF CONTENTS**

1.0 PURPOSE, SCOPE AND APPLICABILITY ..... 1  
 2.0 KEY DEFINITIONS AND ACRONYMS..... 1  
 3.0 SSC SOFTWARE ..... 1  
     3.1 Use and Maintain SSC Software..... 1  
     3.2 Retire SSC Software ..... 6  
 4.0 NON-SSC SOFTWARE ..... 6  
     4.1. Use and Maintain Non-SSC Software..... 6  
     4.2. Retire Non-SSC Software ..... 7

Rev	Date	Description	POC	RM
0	06/23/16	Initial issue	Tobin Oruch, ES-DO	Mel Burnett, CENG-OFF

**1.0 PURPOSE, SCOPE AND APPLICABILITY**

This section describes the “how, when, and who” for using, maintaining and retiring software. See Table 21.6-1 for a summary of this section (for both SSC software and Non-SSC software). See Chapter 21 applicability in SOFT-GEN.

**2.0 KEY DEFINITIONS AND ACRONYMS**

See SOFT-GEN for key definitions and acronyms.

**3.0 SSC SOFTWARE**

**3.1 Use and Maintain SSC Software**

A. How.

1. Control the software to ensure that the use remains consistent with the software design and approved acceptance test results. For ML-1 through ML-3 software, ensure the following controls are in place prior to software approval for use (SWAU). Maintain the software using these controls until the software is retired.
  - a. Application documentation,
  - b. Access control specifications,
  - c. Computer system vulnerability protections,
  - d. Problem reporting and corrective action,
  - e. In-use tests,
  - f. Configuration management,

This search tool may not include the latest files; use individual files for work.

6 Section SOFT-MAINT: Use, Maintain & Retire

Rev. 0, 06/23/16

Table 21.6-1 SOFT-MAINT Summary (This table is a summary only and does not include all requirement details. See text for details.)										
Activity No.	SQM Activity	ML <sup>1</sup>				Implementation Detail			P1040 R9 Ref. Section	ESM Ch. 21 Ref. <sup>5</sup>
		1	2	3	4	How	When	Who <sup>2</sup>		
1	Use and maintain the software	R	R	G	G <sup>3</sup>	<ul style="list-style-type: none"> <li>Use and control the software to ensure that the use remains consistent with the software design and approved acceptance test results</li> <li>Maintain software application documentation</li> <li>Licenses and registrations</li> <li>Use and maintain access controls</li> <li>Use and maintain vulnerability protections</li> <li>Report problems and take corrective actions</li> <li>Perform in-use tests</li> <li>Maintain configuration management (e.g., maintain software baseline, Software Data Sheet [SWDS], and software change documentation)</li> <li>Maintain inventory(ies)<sup>4</sup></li> <li>Implement supporting quality assurance program controls (e.g., training/qualification and assessments)</li> </ul>	<ul style="list-style-type: none"> <li>Upon software approval for use (SWA) and until the software is retired</li> <li>In-use testing and assessments as specified on the SWDS (at a min., when installed on a different computer or after significant changes)</li> </ul>	<ul style="list-style-type: none"> <li>SU (D for use)</li> <li>SRLM (RI for maintenance)</li> </ul>	3.3.7; 3.3.8	SOFT-GEN SOFT-V&V SOFT-INV SOFT-MAINT
2	Retire the software	R	R	R	R	<ul style="list-style-type: none"> <li>Prevent from unintended or routine use</li> <li>Terminate software agreements and/or licenses</li> <li>Remove from the inventory(ies) as applicable<sup>4</sup></li> <li>Where practical and/or where a future need may exist, retain as a record</li> </ul>	<ul style="list-style-type: none"> <li>When no longer required for routine use or (AFU)</li> </ul>	<ul style="list-style-type: none"> <li>SRLM (RI)</li> </ul>	3.3.9	SOFT-INV SOFT-MAINT

<sup>1</sup> ML = Management Level; R = Required; G = Required but graded.  
<sup>2</sup> SU = Software User; SRLM = Software Responsible Line Manager; RI = Responsible individual. The RI may delegate any or all of the work to others, but must retain responsibility for the activity. The SU and SRLM may be the same person.  
<sup>3</sup> In-use testing is not required for ML-4.  
<sup>4</sup> See SOFT-INV for details.  
<sup>5</sup> Ref. = ESM Chapter 21 section name.

This search tool may not include the latest files, use internal files for work.

- g. Inventory, and
  - h. Supporting quality assurance program controls.
2. Process and retain documentation in accordance with the SRLM's governing document control and records management process and/or as described in the SWDS. Ensure the correct SWID (obtained when completing Form 2033) is part of the documentation record number. See [AP-341-402, Engineering Document Management in Operating Facilities](#) for details.
3. Use the following to implement the aforementioned controls. See [IEEE Std 12207, Systems and Software Engineering – Software Life Cycle Processes](#) and [ANSI/ANS-10.8, Non-Real-Time, High-Integrity Software for the Nuclear Industry – User Requirements](#) for additional guidance.
- a. **Software Application Documentation.** Application documentation for SSC software is a general term comprised of the following main categories.
- 1) **Operation and Maintenance (O&M) Documentation.** Software shall be controlled in accordance with approved procedures and instructions (i.e., O&M instructions). They are required for all ML levels. The O&M documentation must be sufficiently detailed and usable to allow a competent individual trained in the use of the software to use the program without undue difficulty and/or likely misuse. *O&M manual(s) should be listed on the SWBL.*
  - 2) **Licenses and Registrations.**<sup>1</sup> As applicable, maintain license documentation and other registrations, including maintenance contracts, as appropriate to promote proper software use. As applicable, document licenses on the SWBL. As applicable, document license and registration maintenance protocols on the SWDS.
  - 3) **Operational Event Documentation.** *Operational event documentation should be retained for ML-1 and ML-2 software to support software maintenance, cybersecurity protection and assessments. Operational event documentation is generated as part of the software operation. It is dependent on the software design. Event documentation may include system startup, shutdown, changes, logons, logon failures, logoffs, self-diagnostic test results, tamper attempts, and/or other information. As applicable, indicate event documentation review and retention protocols on the SWDS.*
  - 4) **Application Logs.** *Application logs are lists (or logs) that provide information that describes who, what baseline revision, when and where (e.g., facility) the software was used. Application logs should be maintained for ML-1 and ML-2, Non-SSC software when such traceability above that provided by the software inventory is required. Application*

---

<sup>1</sup> Licenses and registrations are used to ensure the validity of the software, prevent suspect/counterfeit items (S/CI), and to maintain knowledge of software errors, issues, updates, etc.

*logs may be retained as part of the ES-Div software inventory (possible future capability) or through other means. As applicable, provide application log protocols on the SWDS.*

- b. **Access Control Requirements.** For all ML levels, establish and maintain access controls in accordance with [P218](#), *Cyber Security Access Controls*. Address the security of the computer system and the critical data that may reside on the system. For ML-1 through ML-3 software, document access controls on the SWDS and as required, in the governing software procedures. See SOFT-GEN for SWDS details.
- c. **Computer System Vulnerability Protections.** For all ML levels, implement computer system vulnerability protections as required by [PD210](#), *Cybersecurity Program* and [SD210](#), *Information Risk Management Framework*. Base controls on consequence of loss, confidentiality, integrity, and availability. As required, contact an [Information System Security Officer \(ISSO\)](#) to assist with identifying and implementing computer system vulnerability controls. Consider and as appropriate implement the following protections as part of the O&M procedures and/or as specified on the SWDS:
- 1) Access timeouts after a specified time of inactivity,
  - 2) Recovery/contingency protocols, including retention of backups of system data. (The SWBL serves as the backup for the software, computer programs and documentation needed to run the software).
  - 3) Software tools to verify pedigree authenticity and/or detect tampering associated with suspect/counterfeit items (S/CI).
- d. **Problem Reporting and Corrective Action.** For all ML levels, follow the problem reporting and corrective action processes in SOFT-GEN.
- e. **In-Use Tests.** Perform in-use testing of ML-1 through ML-3 software as specified in this section. In-use testing is not required for ML-4 software unless otherwise specified by the SRLM.
- 1) Perform in accordance with procedures to confirm acceptable performance in the operating environment. Integrate governing SSC work controls.
  - 2) Perform in accordance with the applicable test planning and testing requirements of the SOFT-V&V section.
  - 3) Demonstrate required performance over the range of operation of the controlled function or process.
  - 4) Document test procedures and frequency on the SWDS. Specify test frequency using the following:
    - i. Perform testing after the computer program is installed on a different computer.

- ii. Perform testing when there are significant changes, as determined by the SRLM, in the operating system/environment
  - iii. Perform periodic testing where computer program errors, data errors, computer hardware failures, or instrument drift can affect require performance (consider manual and/or automatic self-check tests).
  - iv. Perform testing when required by procedure (e.g., [AP-341-801](#), Post Modification/Post Maintenance Testing)
  - v. Perform testing based on the risk of failure of the computer program. For ML-1 and ML-2 software, at a minimum, an annual test is recommended.
- f. **Configuration Management.** Perform the following:
- 1) Follow configuration management and interface control processes in SOFT-GEN. Maintain completeness and accuracy of the software baseline and SWDS.
  - 2) Where practical, use and maintain software versions that are compatible with prior versions of the software and associated hardware (sometimes referred to as backward compatibility) to minimize interface issues.
  - 3) As required by the SRLM, ensure data produced by the software is adequately retained (backed up) so that it may be readily retrieved if needed.
- g. **Inventory.** For all ML levels, maintain complete and accurate software inventories in accordance with the SOFT-INV.
- h. **Other Quality Assurance (QA) Controls.**
- 1) Implement other quality assurance (QA) controls in accordance with the governing quality assurance program (e.g., training, assessments, etc.). These other QA controls must be used in conjunction with the software-specific controls of ESM Chapter 21 to provide a comprehensive software QA program. Selected LANL documents that provide controls for the ESM software QA program are listed below for reference:
    - i. [SD330](#), *LANL Quality Assurance Program* (e.g., [P330-3](#), and [P330-6](#), [P330-8](#), [P840-1](#)), and P805, *Export Control*;
    - ii. [P341](#), *Facility Engineering Processes Manual* (e.g., [AP-341-402](#), [AP-341-502](#), [AP-341-507](#), [AP-341-510](#), [AP-341-605](#), [AP-341-504](#), [AP-341-611](#), [AP-341-613](#), [AP-341-703](#), [AP-341-801](#), [AP-341-802](#), [AP-341-901](#));
    - iii. [P342](#), *Engineering Standards* (e.g., [ESM](#) Chapters 1, 7, 8, 12, 15); and,
    - iv. [P343](#), *Facility Engineering Training and Qualification Manual*.



## 6 Section SOFT-MAINT: Use, Maintain &amp; Retire

Rev. 0, 06/23/16

- 2) As applicable, document software-specific training/qualification requirements on the SWDS and complete training/qualification to promote proper use of the software.
  - 3) Document software assessment methods and frequency on the SWDS. Various engineering administrative procedures (e.g., [AP-341-501](#), *Field Walk-down*, *Data Gathering*, and *Inspections*) or LANL processes (e.g., [P330-3](#), *Quality Audits*) may be used to perform software assessments.
- B. When. For ML-4 software, use and maintain software after installation and testing. For ML-1 through ML-3 software, use and maintain after SWAU. For all ML levels, use and maintain until the software is retired. For ML-1 through ML-3 software, perform in-use testing and assessments as specified on the SWDS.
- C. Who. The Software Users (SUs) are responsible for using the software within the software limitations and in accordance with the software O&M requirements. The SRLM is responsible for maintaining the software.

## 3.2 Retire SSC Software

- A. How. Retire software and prevent it from unintended or routine use. Perform the following, as applicable, to retire all ML levels of software:
1. Terminate software support including registrations, agreements and/or licenses.
  2. Remove the software.
  3. Remove safety software from the Institutional-level Safety software inventory per [QPA-DO-FSD-001](#), *LANL Safety Software Inventory Procedure*.
  4. Cancel or revise affected documents per the governing document control process. *Guidance: Also, close all open error and software change requests, regardless of their state.*
  5. Where practical and/or where a future need for the software may exist, retain the retired software baseline (which includes the computer program files) as a record. Retain in accordance with the governing quality assurance records management procedure. Retired software should be retained such that it is retrievable and can be readily installed in the event it is necessary to “rollback” to the software.
- B. When. Retire software when the software is no longer required for routine use and/or when it is no longer approved for use.
- C. Who. The SRLM is responsible for retiring software.

## 4.0 NON-SSC SOFTWARE

## 4.1. Use and Maintain Non-SSC Software

Follow Subsection 3.1 with the following clarifications:

- A. ES-Div: Before using, verify that the software, including correct version, is listed as approved for use (SWAU) on the [ES-Div Website](#). If not listed, contact the SRLM or Engineering Services Software Coordinator (ESSC). Do not use LANL Non-SSC software unless it is listed as SWAU on the ES-Div Non-SSC software inventory website or SWAU is otherwise provided by the SRLM in writing.

**6 Section SOFT-MAINT: Use, Maintain & Retire**

Rev. 0, 06/23/16

- B. ES-Div: Before using, the SU must verify that he/she is authorized to use the software by reviewing the ES-Div software inventory website. If the SU is not shown as an authorized user for the software, contact the SRLM or ESSC. Do not use the software unless authorization is shown on the website or SWAU is otherwise provided by the SRLM in writing.
- C. SSC-related administrative procedures/processes do not apply.
- D. Use the Non-SSC change process in SOFT-GEN (instead of [AP-341-507](#)).

**4.2. Retire Non-SSC Software**

Follow Subsection 3.2 with the following addition:

- A. Remove safety and non-safety software from the [ES-Div Website](#).

*This search tool may not include the latest files; use individual files for work.*