

Contents

1.0 INTRODUCTION.....3

 1.1 Purpose.....3

 1.2 Applicability.....3

2.0 SOFTWARE IDENTIFICATION, OWNERSHIP, AND DETERMINATION3

 2.1 Software Identification and Ownership.....3

 2.2 Management Level Determination.....3

 2.3 Software Summary (optional except for Subcontractors; see SOFT-INTRO).....4

 2.4 Software Determination4

3.0 SOFTWARE DATA SHEET (SWDS; ML-1 through ML-3 only)5

 3.1 Timing5

 3.2 Responsibilities5

4.0 SOFTWARE INVENTORY (ML-1 through ML-4).....5

 4.1 Additional Requirements.....6

5.0 SOFTWARE CONFIGURATION MANAGEMENT.....6

 5.1 Software Baseline (SWBL; ML-1 through ML-3 only).....6

 5.2 Software Change Control8

6.0 PROBLEM REPORTING & CORRECTIVE ACTION10

 6.1 Timing11

 6.2 Responsibilities.....11

7.0 SOFTWARE (PROJECT) RISK MANAGEMENT (GUIDANCE).....11

 7.1 Timing11

 7.2 Responsibilities.....12

8.0 APPENDIX AND ATTACHMENT12

APPENDIX A: CHAPTER 21 SO AND SRLM DECISION DIAGRAM (GUIDANCE).....13

ATTACHMENT 1: CHAPTER 21 SUMMARY TABLE.....15

Record of Revisions

Rev.	Date	Description	POC	RM
0	06/23/2016	Initial issue as provisional document.	Tobin Oruch, ES-DO	Mel Burnett, CENG-OFF
1	05/25/2017	Made chapter mandatory. GEN forms renumbered. Added template 3046, <i>Software Risk Register</i> ; DOE-STD-1073, <i>Configuration Management</i> . Clarified applicability and added Nuclear Criticality Safety, review of supplier software error reports, modified file numbering scheme, Less-Than-Minor Change definition, other minor changes throughout.	Tobin Oruch, ES-DO	Lawrence Goen, ES-DO
2	02/11/2026	Moved some GEN topics into new section, SOFT-INTRO. Clarified ML-4 requirements. Added necessity for new online Form 2033 for all software (excluding LTCC type). Integrated retired inventory section (SOFT-INV deleted). Clarified incorporation of software parameters into software baseline. Revised methods of problem reporting; risk management now guidance. Incorporated VAR-10365R1 (that eliminated AP-341-507) by referencing SSC software change form.	Tobin Oruch, ES-FE	Michael Richardson, ES-DO

Internal users: As with all Los Alamos National Laboratory (LANL) Engineering Standards, please contact the chapter point of contact ([POC](#)) with comments, issues, etc. For LANL, suggestions and questions may be entered in the Engineering Standards Manual (ESM) tool for same [here](#).

SOFT-GEN: General Software Requirements

1.0 INTRODUCTION

1.1 Purpose

SOFT-GEN provides general requirements and information for the implementation of Chapter 21. These general requirements include how to perform the following:

- Identify software and make software determinations
- Implement configuration management (e.g., baselines and software changes)
- Report problems and take corrective actions
- Perform software risk management
- Manage software inventory

1.2 Applicability

See Sections 1.0–2.0 of SOFT-INTRO for a description of the applicability of general Chapter 21 requirements.

Note, for ML-4 “*simple and easily understood*” software that is used in the design of structures, systems, and components (SSCs), and that is individually verified, only selected determination requirements of SOFT-GEN apply (see also that subsection of SOFT-V&V).

2.0 SOFTWARE IDENTIFICATION, OWNERSHIP, AND DETERMINATION

2.1 Software Identification and Ownership

- A. Identify the software owner (SO) and software owner responsible line manager (SRLM) for each piece of software. See Appendix A, *SO, and SRLM Decision Diagram (Guidance)*, for guidance
- B. Analyze SSCs to determine whether there is associated software. Visually inspect and/or review vendor information (e.g., product specifications, catalog data, operations, and maintenance manuals). Identify software as practical and to a level such that software configuration can be controlled. For example, in a control system module that contains multiple pieces of software that cannot be modified other than through replacement of the entire module, identifying the software via the module part number is appropriate.

2.1.1 Responsibilities

The SO is responsible for software identification. During design, the design agency (internal or external) shall normally act for the SO.

SRLMs or divisions should broadcast a software data call on a periodic (at least annual) basis to ensure software is identified and managed as required by [P1040, Software Quality Management](#) and this chapter.

2.2 Management Level Determination

Software management levels (MLs) are used in this chapter to define the level of rigor for applicable software quality assurance requirements. Ensure that a preliminary ML determination has been performed for the software before proceeding.

Section SOFT-GEN: General Software Requirements

Rev. 2, 02/11/2026

See SOFT-INTRO (*subsection 5.0*) for more on software MLs and COE AP-341-502, *Management Level Determinations*, for instructions and details on ML determinations.

2.3 Software Summary (optional except for Subcontractors; see SOFT-INTRO)

This summary aids the user in preparing to fill out [Form 2033](#) in the Graded Software Inventory (GSI). If helpful for project management or for the completion of LANL [Form 2033](#), develop a software summary¹ that includes the following:

- Software name
- Software functional description (i.e., what the software does)
- Software application (i.e., when/where/how the software is used)
- Software management level
- Whether the computer program can be changed other than through replacement of the software (note: replacement includes software upgrades)
- Whether the software is *simple and easily understood* software that will be individually verified each time the software is used

Resources on the SOFT-GEN section of the *Engineering Standards Manual* (ESM) webpage include a software summary template.

2.3.1 Timing

For non-SSC software, provide the software summary to the LANL SRLM at the earliest practicable time and before software acquisition or design (or use, if the software is already on hand).

For SSC software, provide draft software summaries no later than the 60% project design deliverables. When a required field on the software summary is not known, indicate TBD. Deliver the final summary (complete without TBDs) with the 90% detailed design deliverables.

2.4 Software Determination

Except for ML-4 “*simple and easily understood*” software that is used in the design of structures, systems, and components (SSCs), and that is individually verified (Requirement 21-0201):

- A. Obtain a software identification number (SWID) for each piece of software that will follow Chapter 21. A link to the SharePoint numbering utility is on the Chapter 21 webpage.

Note: Separate but related software elements may be identified and subsequently controlled and inventoried (as applicable) as a software collection. For example, there may be various software elements—control system executable, human-machine interface (HMI) software, field element embedded software, etc.—associated with a gas handling process control system software. The software may be identified, controlled and inventoried as one item, the gas-handling process control system software.

- B. Complete a [Form 2033](#) using the graded software inventory process.

2.4.1 Timing

Complete [Form 2033](#) in the graded software inventory at the earliest practical time and before software acquisition or final design. For SSC software, submit draft software grading information with the 60% detail design

¹ The software summary 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.

Section SOFT-GEN: General Software Requirements

Rev. 2, 02/11/2026

submittal. When a required field on the form is not known, indicate TBD or similar term. Submit complete forms (without TBDs) with the 90% detailed design submittals.

2.4.2 Responsibilities

The SO is responsible for the completion and submittal of [Form 2033](#). During design, the design agency (internal or external) shall normally act for the SO.

3.0 SOFTWARE DATA SHEET (SWDS; ML-1 through ML-3 only)

Prepare a Software Data Sheet (SWDS). This document addresses general topics such as planning, managing, access control, and training (Requirement 21-0202; INTRO Requirement 21-0107). See SOFT-GEN-FM01, *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., EDRMS and AP-341-402, *Engineering Document Management in Operating Facilities*). If the SWDS references/relies entirely on other documents for actual control during the use phase, they must be maintained/revised while the SWDS need not be.

Notes: For ML-4, data sheets are not required; however, they may be used at the SRLM's discretion.

The initial revision of the SWDS may have TBD entered for some data (e.g., fields in the Use and Maintenance section) that are not known during initial planning.

If a training and qualification plan attached to (or separate from and referenced by) the SWDS is desired, [Form 2331 - Software Training and Qualification Plan \(SWTQP\)](#) is available.

3.1 Timing

Complete a SWDS at the earliest practical time and before software acquisition or final design. For SSC software, submit a draft SWDS with the 60% detailed design submittal unless deferring to construction phase. When a required field on the form is not known, indicate TBD. Provide complete forms (without TBDs) with the 90% detailed design submittals.

3.2 Responsibilities

The SO is responsible for the completion and submittal of the SWDS. During design, the design agency (internal or external) will normally act as the SO.

4.0 SOFTWARE INVENTORY (ML-1 through ML-4)

This subsection describes the steps involved in inventorying software throughout its life cycle. (Requirement 21-0203)

Note: Do not enter classified or controlled unclassified information (CUI) into the GSI (or any other system not authorized for same). The Graded Software Inventory (GSI) is the primary tool for inventorying software at LANL and implements this function via approved Form 2033s. Organizations may implement supplemental software inventories if desired.

- Ensure any required software inventory activities are complete no later than approval for use (SWAU).
- Update the software status and inventory data in the GSI as needed thereafter to ensure completeness and accuracy.
- Update the software status in the GSI upon retirement of software or upon cancellation of the associated projects (if the projects did not go to completion).
- See also Section SOFT-MAINT for inventory expectations, especially for FAC-COE division control of design software.

4.1 Additional Requirements

4.1.1 Subcontractor-Provided Software

Subcontractors that provide ML-1, -2 or -3 SSC software to LANL must provide the minimum required SSC software inventory information by SOFT-INTRO (§6.2) and -GEN as part of the project equipment list (PEL) or via other submittal per ESM Chapter 1, Section Z10.

5.0 SOFTWARE CONFIGURATION MANAGEMENT

Maintain software configuration management (SCM) in accordance with this section. This section includes the development of a baseline of configuration-controlled items and the change control processes utilized to make es-controlled changes to the baseline. (Requirement 21-0204)

5.1 Software Baseline (SWBL; ML-1 through ML-3 only)

A baseline 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). In the event the in-use computer program fails, becomes corrupted, or other problems occur, the baseline may be used to re-establish operation. (Requirement 21-0205)

Note: Software baselines are not required for ML-4 software; however, they may be used at the SRLM's discretion.

- A. Complete SOFT-GEN-FM02, Software Baseline Form (SWBL) by identifying the following as applicable:
- 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.
 - Computer program(s): source, object, back-up, data, configuration, and support software files needed to run the program. Configuration files include limits, settings, bounding set points, etc. As applicable, include the file/directory attributes and authorities. These are the required access settings for reading, writing and executing for users, owners, etc.
 - Associated hardware: identify the associated hardware needed to run the computer program in the system design description document(s) or directly in the baseline.

For SSC Software

- Ensure the software baseline is consistent with the SSC technical baseline documents, as applicable. 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 in Operating Facilities*.
- Ensure that the software baseline includes any configuration data and/or parameters required for the component to perform its function(s). The document containing this information must
 - Clearly identify any configuration data or parameters that act as set points for safety class (SC) and safety significant (SS) implementation of safety actions in hazard category 2 or 3 nuclear facilities, and which are therefore subject to the requirements of AP-341-613, *Instrumentation Set Point Control*.
 - Be reviewed, approved, and signed by the SO and SRLM.

Priority should be placed on recording critical parameters, including set points, that are normally entered or modified via a user interface on the SSC. If SSC configuration data and/or parameters are contained in a file or profile that is uploaded to the SSC (or associated control system), it is not necessary to define them separately as long as the associated file(s)

are included in the software baseline. SSC software parameter values may be assigned an allowable range (e.g., air compressor loading pressure may be set between 89-95 psig) as long as component functionality is assured at all values within the range.² Parameters may be changed within the defined allowable range without engaging in the software change control process. The documentation may define multiple SSC configurations (i.e., normal operation / test and balance activities / emergency override).

- B. Control the software baseline documentation in accordance with the SRLM's document/electronic document control process (e.g., EDRMS). The SRLM's approved document control process is used, as a minimum, to control both the documents and the computer program files that comprise the baseline. The document control process must ensure only authorized access and changes are made to software baseline documentation. Ensure the correct SWID is part of the SWBL record number and subsequent software documentation.
- C. Update the software baseline after software changes. Ensure that the baseline defines the most recent approved software configuration. Ensure that design and/or analysis work is performed on computers, servers, etc. using documentation and computer programs that match (mirror) the baseline as described and controlled in the SRLM's document control process.
- D. When support software (including software tools) is used for new or modified software, one must evaluate, review, accept it for use, and place it under configuration control. 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. (Requirement 21-0206)
- E. Implement a baseline labeling system in accordance with the following (Requirement 21-0207):³
 - Identify (uniquely) each configuration item (documentation or computer program).
 - Identify changes to configuration items by revision.
 - Provide the ability to uniquely identify each configuration of the revised software available for use.
 - Use the SRLM's approved document numbering system for software baseline file and/or document labeling. If there is none, consider utilizing the CoE system described by AP-341-402, *Engineering Document Management in Operating Facilities*.
 - Include baseline label information within the computer source code where possible.

Examples

 - 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., 1.1, 1.2...) and the SWCP number driving the change.
 - Use the COTS or existing software labeling configuration for commercial off-the-shelf software (COTS) or other (existing) software with a compliant labeling configuration. For other software, use Table GEN-1, *Computer Program File Labeling System*, or an alternative as approved by the SLRM.
 - Ensure the correct SWID is integrated into the baseline labeling where possible.

² For parameters assigned a range, the currently implemented values should be recorded in an equipment logbook or other document.

³ Except where the LANL SRLM determines that doing so is not possible or practical, but CM is nevertheless achieved.

Table GEN-1 Preferred Computer Program File Labeling System			
Prior to Approval for Use ¹		After Approval for Use	
Major Change ²	Minor Change ²	Major Change	Minor Change
A, B, C...	A.1, A.2, A.3...	1, 2, 3...	1.1, 1.2, 1.3...
¹ 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. ² See definitions of Major Change and Minor Change in SOFT-INTRO App A, Definitions.			

5.2 Software Change Control

- A. Follow Table GEN-2 (for all ML levels) and then, when required by the table, apply the following requirements to major and minor software changes (Requirement 21-0208):
 - 1. For designed software, institute design control measures commensurate with those applied to the original design.
 - 2. Document changes and include the following:
 - A description of the change;
 - The rationale for the change;
 - Identification of affected software baselines, including documents to be updated; and,
 - Evaluation and approval of the change (see SOFT-V&V for evaluation methods/criteria).

Table GEN-2 Software Change Summary by Software Management Level (ML)									
(This is a summary only and does not include all the requirement details in the text. See text for details.)									
For the following types of software changes	Provide the following deliverable	SSC Software ML ¹				Non-SSC Software ML ¹			
		1	2	3	4	1	2	3	4
Any change made <u>before or after</u> approval for use (SWAU). Includes Less than Minor, Minor, and Major changes.	Documented comment in the computer program code where feasible	R	R	R	R	R	R	R	R
Minor and Major Changes made during design implementation (e.g., fielding or startup) and <u>before</u> SWAU	AP-341-519 change documents ²	R	R	Gr	Gr	-	-	-	-
Minor and Major Changes made once software is baselined	Revised baseline ³	R	R	R	-	R	R	R	-
Minor and Major Changes made <u>after</u> SWAU	SSC: SSC Software Change Package Form (SWCP) SOFT-GEN-FM04 ⁴	R	R	Gr	-	-	-	-	-
	Non-SSC: SWNCP Form, SOFT-GEN-FM03	-	-	-	-	R	R	Gr	-

Table GEN-2 Software Change Summary by Software Management Level (ML) (This is a summary only and does not include all the requirement details in the text. See text for details.)									
For the following types of software changes	Provide the following deliverable	SSC Software				Non-SSC Software			
		ML ¹				ML ¹			
		1	2	3	4	1	2	3	4
<p>¹ See Form 2033 for software type and software risk levels. R = Required. Gr = Required but graded. “-“ = Not required.</p> <p>² Includes Field Change Notices (FCNs), Field Change Requests (FCRs), and Design Revision Notices (DRNs) as described in AP-341-519 unless explicitly superseded by another administrative procedure.</p> <p>³ Use SOFT-GEN-FM02, <i>Software Baseline Form (SWBL)</i>. Include documents and computer program files.</p> <p>⁴ See SOFT-V&V; ESM Chapter 15, <i>Commissioning</i>; and AP-341-801, <i>Post Modification/Post Maintenance Testing</i> for related testing requirements of SSC software changes.</p>									

- B. Perform appropriate verification activities and validation (V&V) for software changes (see SOFT-V&V).
- C. When making ML-1 or -2 computer program changes (revisions) through acquisition, such QL-1 items/services from suppliers not on the IESL must be commercial-grade dedicated (CGD). Use the following graded approach for dedicating software changes unless a more conservative approach is required by the LANL SRLM:
 - Dedicate major changes (see Definitions).
 - 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.
- D. Incorporate the changes in documentation and maintain traceability of the change to the software design requirements.

Include the following:

 - Initiation, evaluation, and disposition of a change request
 - Control and approval of changes prior to implementation, including USQ when applicable
 - Requirements for retesting (e.g., regression testing) and acceptance of the test results
- E. Perform the following tasks:
 - Maintain the status of configuration items and control configuration changes until they are incorporated into the approved product baseline.
 - Maintain the status of changes that are proposed and approved but not implemented.
 - Provide notification of this information to affected organizations.
 - Use the governing document control process (for changes to documentation only).
 - Implement the processes described and/or referenced in this chapter to ensure that interfaces are controlled/managed such that unintended consequences do not occur.
 - Update the software baseline (if required).
- F. Retire and archive software when no longer needed (per SOFT-MAINT).

5.2.1 Less Than Minor Computer Program Changes

Ensure that less than minor computer program changes are made by a competent individual knowledgeable in the software. If feasible, add a brief description, the name of who is making the change, a justification that the change does not modify performance functions (all MLs), and the date that the less than minor change is made in the computer program code at the time the change is made. If it is not feasible to document the less than minor change in the code, then document in associated operational logbooks, etc.

5.2.2 Responsibilities

- 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.
- As determined by the SRLM, only those who are knowledgeable in the computer program code may make changes to the computer program code.

6.0 PROBLEM REPORTING & CORRECTIVE ACTION

- A. Problem reporting and corrective action (PR&CA) must be used for in-use (production) software to document, evaluate, and correct software problems. A software problem (i.e., error) is defined as any condition deviating from an established baseline, including deviations from the current approved software baseline. (Requirement 21-0209).
- B. 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 or trac), or formal methods (see below) may be used throughout the software lifecycle.

Note: [Form 2326 - Software Problem Report \(SPR\)](#) may be used where desired by SRLM.

- C. For ML-1 through ML-3 software, before SWAU, less formal methods may be used. After SWAU, formal methods must be used. Individually or in combination, P330-6, *Nonconformance Control and Reporting*, P322-4, *Issues Management*, or an equivalent process (as determined by the SRLM) must be used, especially when a problem is determined to affect the operability of the software or the associated SSC.⁴ Regardless, it must address the following:
 - Describe the evaluation process for determining whether a reported problem is an error (see definitions) or other type of problem (e.g., user mistake).
 - Define the responsibilities for disposition of problem reports, including notification to the originator of the results of the evaluation.
 - When a problem is determined to be an error, provide the following information (as appropriate):
 - How the error relates to appropriate software engineering elements
 - How the error impacts past and present use of the computer program
 - How the corrective action impacts previous development activities
 - How the users are notified of the identified error, its impact, and how to avoid the error, pending implementation of corrective actions
 - Include methods for documenting, evaluating, and correcting software problems. This should include an estimate of the level of effort and corrected software release date.
 - Promptly identify conditions adverse to quality and correct them as soon as practicable.

⁴ For SSC software: Other, more immediate processes such as AP-341-516 Operability Determination, may also be appropriate. Prompt initiation and implementation of a DCF is an alternative to P322-4, with FDAR approval.

- In the case of a significant condition adverse to quality, determine the cause of the condition and take corrective action to preclude recurrence.
- Document the identification, cause, and corrective action for significant conditions adverse to quality and report this information to appropriate levels of management.
- Verify that corrective actions are completed.
- When software vendors and/or suppliers provide notifications of errors, review such notifications at a frequency commensurate with the software risk level and frequency of use. A review of problem reports prior to use in calculations and/or a minimum of once per year is recommended.

6.1 Timing

Perform problem reporting and corrective action throughout the software life cycle.

6.2 Responsibilities

The SO performs software PR&CA on behalf of the SRLM. The SRLM is responsible for ensuring PR&CA is performed as required by this chapter and associated procedure(s), e.g., P330-6.

7.0 SOFTWARE (PROJECT) RISK MANAGEMENT (GUIDANCE)

Software risk management is a good practice that focuses on the risks to successful completion of a software project; it does not focus on the risks of potential failure of the software. Software risk management applies to all phases of the software project life cycle. Overall software risk management is achieved through implementation of the requirements-based, systematic processes of this chapter and the supporting institutional quality assurance program. Successful implementation of this chapter should result in successful completion of a software project.

- A. For ML-1 through ML-3 software, SRLM should strongly consider addressing software-specific risks (risks specific or unique to the software implementation). Manage software-specific risks by (a) assessing and (b) controlling the risks.
 - Utilizing a graded approach approved by the SRLM and chapter POC, implement risk management based on industry accepted methods such as those described in DOE SQAS21.01.00 1999, *Software Risk Management – A Practical Guide*; DOE G 413.3-7, *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*.
- B. For ML-4 software, the SRLM should determine if project-level software risk management is needed (e.g., complex, large and/or important projects)
- C. Include risk identification, analysis, and prioritization in the risk management process to ensure that the necessary resources are available to mitigate risks. Additionally, include risk resolution and tracking.
- D. Resolve risks using risk avoidance, mitigation, and/or transference.
- E. Document and maintain software-specific risks on a risk register (also referred to as risk list) in the SWDS or, if desired, in a separate document. Attaching the risk register to the SWDS is the best practice. If not attaching it to the SWDS, create a unique software risk register (SWRR) number; many organizations obtain that [here](#).

7.1 Timing

Projects should develop a SWRR at the earliest practical time (before design/acquisition as applicable), and perform risk management throughout the software life cycle (e.g., review periodically).

7.2 Responsibilities

The SO performs software risk management on behalf of the SRLM. The SRLM should ensure it is performed, and review and approve the risk register.

8.0 APPENDIX AND ATTACHMENT

- Appendix A, Chapter 21 SO and SRLM Decision Diagram (Guidance)
- Attachment 1, Chapter 21 Summary Table

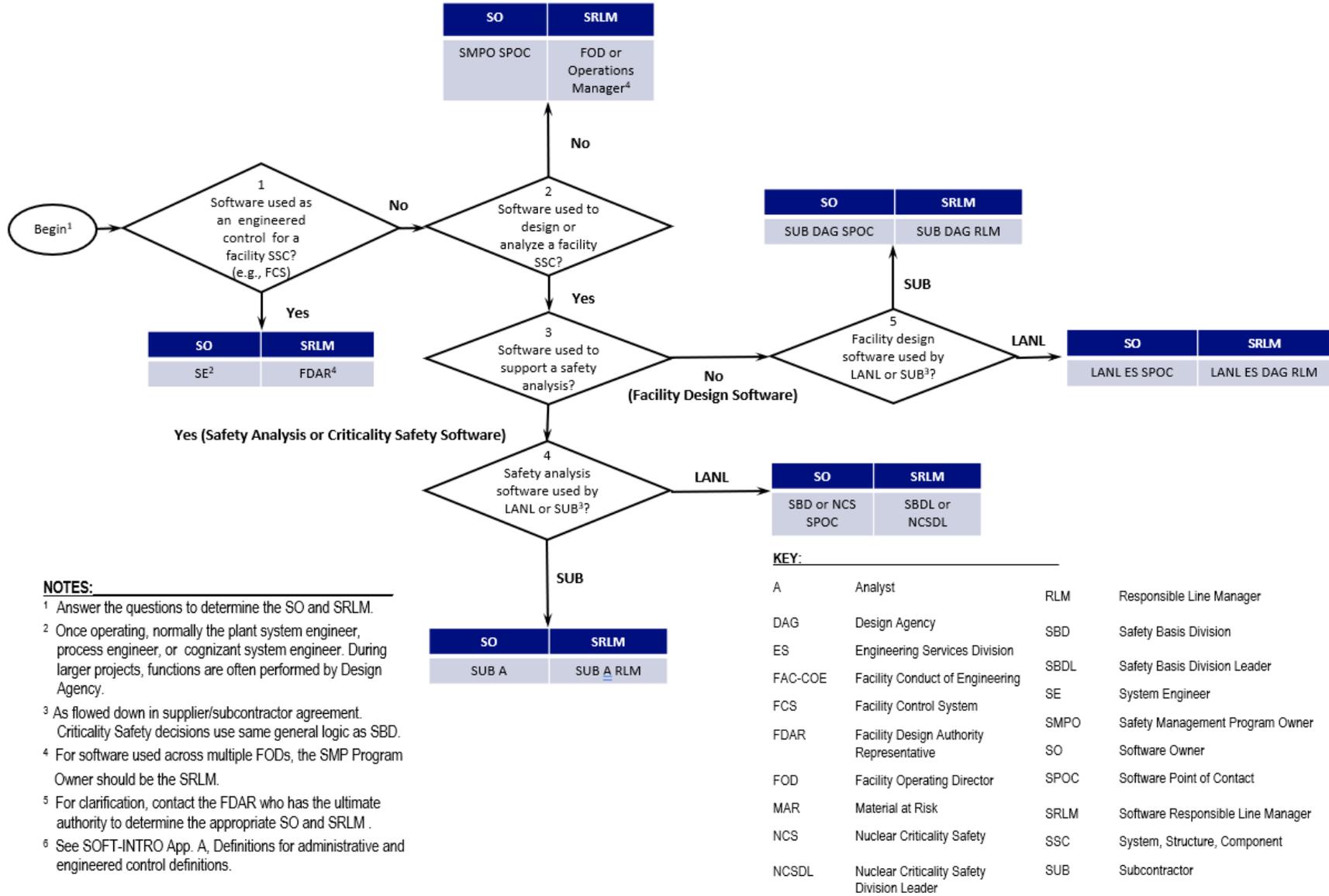
APPENDIX A: CHAPTER 21 SO AND SRLM DECISION DIAGRAM (GUIDANCE)

Use Figure 21.A1 to identify the SO and SRLM. For clarification, contact the Facility Design Authority Representative (FDAR).⁵ .

- 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.
- For fire alarm control panel (FACP) software, the system engineer is the SO. The Fire Protection group provides support to the SO to promote chapter compliance.
- For SB and NCS divisions, Chapter 21 FDAR responsibilities shall be performed by the Senior Director of the Nuclear Safety Program (ALDFO) or the TA-55 Chief Engineer. This includes approving forms and commercial grade dedication documents (incorporates/supersedes [VAR-10326r1](#)).

⁵ Note: Chapter 21 involvement of FDARs is based on roles defined in NQA-1 and DOE-STD-1073 (and former DOE O 414.1D, that stated: "Management of safety software must include the following elements. Involve the facility design authority, as applicable, in: the identification of; requirements specification; acquisition; design; development; verification and validation (including inspection and testing); configuration management; maintenance; and retirement.")

Figure 21.A1 Chapter 21 SO and SRLM Decision Diagram (Guidance)



NOTES:

- ¹ Answer the questions to determine the SO and SRLM.
- ² Once operating, normally the plant system engineer, process engineer, or cognizant system engineer. During larger projects, functions are often performed by Design Agency.
- ³ As flowed down in supplier/subcontractor agreement. Criticality Safety decisions use same general logic as SBD.
- ⁴ For software used across multiple FODs, the SMP Program Owner should be the SRLM.
- ⁵ For clarification, contact the FDAR who has the ultimate authority to determine the appropriate SO and SRLM.
- ⁶ See SOFT-INTRO App. A, Definitions for administrative and engineered control definitions.

ATTACHMENT 1: CHAPTER 21 SUMMARY TABLE

Rev	Date	Description	POC
0	06/23/16	Initial issue	Tobin Oruch, ES-DO
1	05/25/17	Updated for Rev. 1 of chapter	Tobin Oruch, ES-DO
2	02/11/26	Updated for Rev. 2 of chapter	Tobin Oruch, ES-FE

This table is summary of the entire chapter (less SOFT-INTRO), addressing both SSC software and Non-SSC software. It is largely synchronized with the requirements in each Section; however:

- Sections have text and Section-specific tables that are more detailed, and
- In the case of a conflict, contact the Chapter POC (Section requirements will generally supersede those in this summary).

Table 21.ATT 1 Chapter 21 Summary by Software Management Level (ML)								
This table is a summary only (does not include all requirement details, nor much of SOFT-INTRO). See text of Section(s) for details.								
Section Name/ Subsection ¹	Activity Description	Deliverable	SSC Software ML ²			Non-SSC Software ML ²		
			1, 2	3	4	1, 2	3	4
SOFT-GEN, 2	Identify software, evaluate and complete software determination	Software summary (only required for subcontractors; ref. SOFT-INTRO, §6.2.A)	R	R	R	R	R	R
		Form 2033	R	R	Gr ³	R	R	Gr ³
SOFT-GEN, 3	Plan software-specific activities	Software data sheet form SOFT-GEN-FM01 (SWDS)	R	R	-	R	R	-
SOFT-GEN, 4	Inventory SSC software	Updated Form 2033/graded software inventory	R	R	-	-	-	-
SOFT-GEN, 4	Inventory non-SSC Facility Eng software (e.g., ES, PFE, PIE)	Updated Form 2033/graded software inventory; see SOFT-MAINT §4.1	-	-	-	R	R	R
SOFT-GEN, 5.1	Develop and maintain a software baseline (SWBL)	Software baseline form SOFT-GEN-FM02 (SWBL) ⁴	R	R	-	R	R	-
SOFT-GEN, 5	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
	Minor or major changes made <u>before</u> SWAU	AP-341-519 change documents ⁵	R	Gr	Gr	-	-	-
	Minor or major changes made once software is baselined	Revised SWBL ⁴	R	R	-	R	R	-
	Minor or major changes made <u>after</u> SWAU	SSC Software Change Form SOFT-GEN-FM04	R	Gr	-	-	-	-
Non-SSC Software Change Form SOFT-GEN-FM03 (SWNCP)		-	-	-	R	Gr	-	
SOFT-GEN, 6	Report problems and take corrective action (PR&CA):							
	Before SWAU	bug lists and/or documented comment field in the computer program	R	R	Gr	R	R	Gr

Table 21.ATT 1 Chapter 21 Summary by Software Management Level (ML)								
This table is a summary only (does not include all requirement details, nor much of SOFT-INTRO). See text of Section(s) for details.								
Section Name/ Subsection ¹	Activity Description	Deliverable	SSC Software ML ²			Non-SSC Software ML ²		
			1, 2	3	4	1, 2	3	4
	After SWAU	P330-6 (Nonconformance Reports [NCRs]), P322-4 (IM), other methods.	R	R	Gr	R	R	Gr
SOFT-GEN, 7	Manage software (project) risks	Risk register (SWRR or list) as part of SWDS	-	-	_ ¹²	-	-	_ ¹²
SOFT-ACQUIRE, 2.1	Acquire existing LANL software if suitable	Update authorized users/SWDS as required; generate an interface agreement where desired	-	-	-	R	R	Gr
SOFT-ACQUIRE, 2.1, 2.2	Acquire new software	▪ Statement of Work (SOW)	R	R	-	R	R	-
		▪ SWDS	R	R	-	R	R	-
		▪ Commercial Grade Dedication (CGD) Documentation per AP-341-703	R ⁶	R ⁶	-	R ⁶	R ⁶	-
		▪ P-card/Purchase Order (PO)	R	R	R ⁷	R	R	R ⁷
		▪ Exhibit H	R	R	R ⁸	R	R	R ⁸
		▪ Inter-/Intra- LANL Agreements ⁹	R ⁹	R ⁹	R ⁹	R ⁹	R ⁹	R ⁹
SOFT-DESIGN, 3	Develop software design requirements (design input)	▪ Software design inputs: <ul style="list-style-type: none"> ○ SSC drawings ○ SSC performance specification ○ FDD or SDD 	R	Gr	Gr	-	-	-
		▪ Other SSC design inputs/outputs as required						
		▪ Software system requirements specification (SWYRS)	--	-	-	R	Gr	_ ¹³

Table 21.ATT 1 Chapter 21 Summary by Software Management Level (ML)								
This table is a summary only (does not include all requirement details, nor much of SOFT-INTRO). See text of Section(s) for details.								
Section Name/ Subsection ¹	Activity Description	Deliverable	SSC Software ML ²			Non-SSC Software ML ²		
			1, 2	3	4	1, 2	3	4
SOFT-DESIGN, 4	Design the software (design output)	<ul style="list-style-type: none"> ▪ Software design outputs: <ul style="list-style-type: none"> ○ software specification (SWRS) ○ software design (SWDD) ○ Software system hazard analysis and mitigation documentation (SWHA; optional) ○ software design traceability documentation (SWTM) ○ SOFT-GEN deliverables ○ Other software design deliverables as required ○ Operations and maintenance (O&M) instructions ○ computer model mathematical terms¹⁰ 	R	Gr	Gr	R	Gr	- ¹³
SOFT-DESIGN, 5	Translate the software design into computer program language (code)	<ul style="list-style-type: none"> ▪ Computer program language (code): <ul style="list-style-type: none"> ○ data structures/files, ○ source code (where feasible), ○ executable code ▪ Computer program listings (e.g., pdf printout of source code) 	R	Gr	Gr	R	Gr	Gr
SOFT-V&V, 4.1	Review (input) requirements	<ul style="list-style-type: none"> ▪ Dispositioned review comments 	R	Gr	Gr	R	Gr	Gr
SOFT-V&V, 5.1	Review acquisition	<ul style="list-style-type: none"> ▪ Acquisition review document 	-	-	-	R	R	-
SOFT-V&V, 5.2	Develop and review the test plan (SWTP)	<ul style="list-style-type: none"> ▪ Test plan (e.g., Form 3055) ▪ Dispositioned review comments 	R	Gr	Gr	R	Gr	Gr
SOFT-V&V, 4.3	Review the software design (SWDD)	<ul style="list-style-type: none"> ▪ Software design ▪ Dispositioned review comments 	R	Gr	Gr	R	Gr	Gr
SOFT-V&V, 4.4	Review computer program code where feasible	<ul style="list-style-type: none"> ▪ Dispositioned code review comments 	R	-	-	R	-	-

Table 21.ATT 1 Chapter 21 Summary by Software Management Level (ML)								
This table is a summary only (does not include all requirement details, nor much of SOFT-INTRO). See text of Section(s) for details.								
Section Name/ Subsection ¹	Activity Description	Deliverable	SSC Software ML ²			Non-SSC Software ML ²		
			1, 2	3	4	1, 2	3	4
SOFT-V&V, 4.5	Test and review test results	<ul style="list-style-type: none"> ▪ Interim tests and test (qualification) reports ▪ Simulated acceptance test and acceptance test report ▪ Acceptance test and test (qualification) report (SWTR) ▪ Dispositioned review comments 	R	Gr	Gr	R	Gr	Gr
SOFT-V&V, 5.5	Review and approve software for use (SWAU)	<ul style="list-style-type: none"> ▪ SWAU request package ▪ Dispositioned review comments ▪ Software approval for use form SOFT-V&V-FM01, SWAU 	R	Gr	-	R	Gr	-
SOFT-MAINT, 3.1, 4.1	Use and maintain the software	<ul style="list-style-type: none"> ▪ Operation and maintenance (O&M) documentation ▪ Operational event documentation ▪ Application logs ▪ Licenses and registrations ▪ Access controls ▪ Computer system vulnerability protections ▪ Problem reports and corrective actions per SOFT-GEN ▪ In-use test plans, tests, review documentation, and test reports per SOFT-V&V ▪ Updated, configuration management, SWDS and deliverables as described on SWDS and SWBL per SOFT-GEN ▪ Accurate software inventory per SOFT-INV ▪ Other quality assurance control documentation (e.g., training records, assessments) 	R	Gr	Gr	R	Gr	Gr
SOFT-MAINT, 3.2, 4.2	Retire the software	<ul style="list-style-type: none"> ▪ Software removed to prevent unintended or routine use ▪ Terminated licenses/agreements ▪ Updated software inventory ▪ Cancelled or revised affected documents ▪ Software record 	R	R	R	R	R	R

Notes:

- ¹ ESM Chapter 21 section name and subsection number.
- ² See Form 2033 for software type and SRL. R = Required. Gr = Required but graded. "-" = Not required.
- ³ For ML-4 SSC software, complete Form 2033 if there is a reasonable probability that it could be ML-1, ML-2, or ML-3 software, or if the highest function of the SSC is greater than ML-4. For Non-SSC ML-4, 2033 not required when "simple and easily understood" and each-time verified.
- ⁴ See SOFT-GEN-FM02, *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.
- ⁵ 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.
- ⁶ Required only for ML-1 or ML-2 nuclear safety software that is not procured from an NQA-1 qualified supplier.
- ⁷ A PO or a Purchase Card (P-card) can be used for ML-4 software; a P-card cannot be used for ML-1 through ML-3 software.
- ⁸ Required if the supplier comes onsite (at LANL) to install and/or maintain software.
- ⁹ Required, or should be required, if software is acquired from LANL organizations, DOE sites or government agencies.
- ¹⁰ Required for computer program models only.
- ¹¹ deleted
- ¹² For risk management, following Chapter 21 without specific attention to this is adequate for all but the most complex/critical designed software efforts.
- ¹³ For "Simple and Easily Understood Non-SSC," designed, ML-4 software, the V&V may performed "each-time" by the associated calculation (e.g., per AP-341-605) rather than typical, pre-verified alternative.