Section SOFT-INTRO: Chapter Introduction Rev. 2.1, 03/16/2026

Contents

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, *Software Risk Register*; DOE-STD-1073, *Configuration Management*. 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	New section SOFT-INTRO created for introductory material from GEN. Incorporated VAR-10554 allowing alternative use of P1040 and other applicability changes regarding LTCC, firmware, PD340, R6, pre-existing software, subcontractor responsibilities, and FDAR role fulfillment for Safety Basis and Nuclear Criticality Safety (superseding VAR-10326r1). Clarified requirement for safety software testers to be qualified. In all Sections, requirements correlated to bases in Requirements ID file (LANL-use); general updates including references.	Tobin Oruch, *ES-FE*	Michael Richardson, *ES-DO*
2.1	03/16/2026	Subsection 2.3 exclusions expanded/clarified.	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 (POCs) with comments, issues, etc. For LANL, suggestions and questions may be entered in the Engineering Standards Manual (ESM) tool for same here.

NOTE: Because of the extent of Revision 2 (and 2.1), if clarification to a user is warranted due to conflict, ambiguity, or omission, the Chapter POC alone may issue written direction when such cannot diminish nuclear safety and only involves use of Revision 1 approach.

SOFT-INTRO: Chapter Introduction

1.0 INTRODUCTION

1.1 Purpose

Chapter 21 provides the default software quality assurance (SQA) plan for software within the programs listed in Subsection 2.1 below. Following this plan, together with the application-specific implementing documents required by this chapter, provides reasonable assurance that in-scope software will consistently, compliantly, and efficiently perform its intended function(s). SQA requirements described in this chapter are derived from the following mandates:

- Department of Energy (DOE) Order 414.1E, *Quality Assurance*;
- ASME NQA-1-2008/NQA-1A-2009*, Quality Assurance Requirements for Nuclear Facility Applications*, Parts I and II (hereafter NQA-1) (Requirement 21-0101);
- DOE-STD-1073-2016, *Configuration Management*; and
- LANL PD330, *LANL Quality Assurance Program Description*, and related P330-X policy.

Figure 21.1 shows the flow down of SQA requirements from the LANL Prime Contract to Chapter 21.

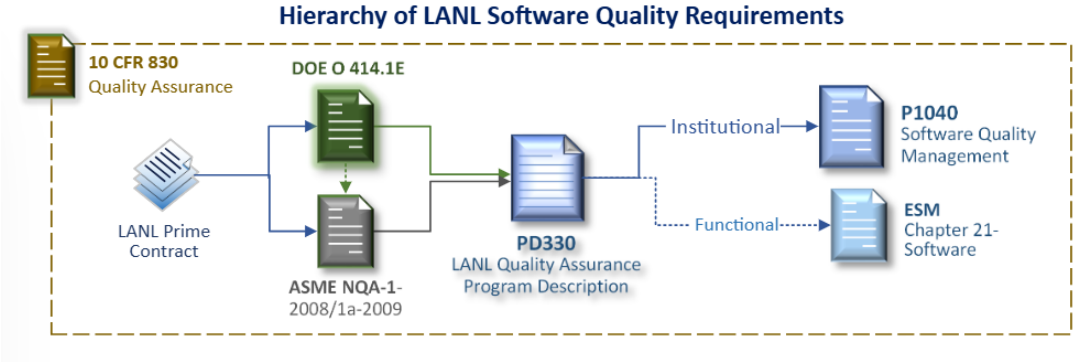


Figure 21.1 Flow Down of Software Quality Requirements

1.2 Chapter Usage

- (Requirement 21-00YZ): When this phrase appears herein, it is a LANL-internal reference to a basis file that captures and categorizes LANL Standards Manual drivers and is not relevant to most users.
- Fonts: Where appropriate throughout the ESM, guidance is provided to aid in the implementation of requirements. Guidance will be in *italicized text* and/or otherwise clearly indicated (e.g., by headings). Document titles or defined terms in italics are an exception. All other text (plain type) indicates mandatory requirements, as does “shall” or “must”.
- Appendix A of SOFT-INTRO contains a list of defined Chapter 21 terms and abbreviations. Throughout the chapter, defined terms are often capitalized, italicized, or set in quotes.

- Appendix B is a list of references related to Chapter 21.
- For an "underway" project, when the Tailored Standards Manual is in the code of record rather than the ESM, take Chapter 21 references to other ESM Chapters to mean the corresponding TSM chapter.

2.0 APPLICABILITY

2.1 General

Chapter 21 is the default approach to software quality assurance (SQA) within the scope of the Facility Conduct of Engineering, Safety Basis, and Nuclear Criticality Safety programs[1], specifically[2]:

- PD340, *Conduct of Engineering and Configuration Management for Facility Work*;
- PD110, *Safety Basis*; and,
- SD130, *Nuclear Criticality Safety Program*. (Requirement 21-0102)

In accordance with PD340, this chapter is therefore also applicable to programmatic and/or R&D engineering software that affects structures, systems, or components (SSCs) described in a Documented Safety Analysis (DSA)[3], or associated with collocated SSCs.[4]

Regarding P1040: Instead of using Chapter 21, software owners may instead utilize P1040, *Software Quality Management* with the approval of the software responsible line manager (SRLM) or Facility Design Authority Representative (FDAR).[5] This chapter and P1040 are both acceptable LANL SQA programs; it is not necessary to simultaneously satisfy the requirements of both.

> *P1040 has more detail on SQA planning, while this chapter is the preferred method for SQA for the programs above, especially PD340 work, since it is integrated with such.*
>
> *Generally, software not included in the above three programs is subject instead to the requirements of P1040.*

Guidance: For ML-1, ML-2, and some ML-3 SSC systems for which software operational complexity and life cycle maintenance can be disadvantageous, evaluate and, as appropriate, employ designs that use no or minimal software.

2.2 Software Types

Chapter 21 defines and applies to two major types of software:

- Non-SSC software (see definitions), including "Simple and Easily Understood" software used in the design of SSCs, and
- Structure, system, and component (SSC) software
  - This includes Read-only/Firmware/Embedded software, which is defined as the combination of a hardware device and computer programs and data that reside as read-

[1] I.e., SB and NCS division SQA, but not every program by others that supports these safety management programs. Also, for these 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).
[2] At the time of writing, the programs include the activities of these divisions (and PD340-scope activities of several others): Engineering Services (ES), Plutonium Facilities Engineering (PFE), Plutonium Infrastructure Engineering (PIE), Safety Basis (SB), and Nuclear Criticality Safety (NCS).
[3] The specific SSCs cited in the DSA subject to this requirement are nuclear safety SSCs (Safety Class and Safety Significant SSCs), Other Hazard Controls (OHCs), and/or other SSCs that are listed as providing defense-in-depth.
[4] Collocated SSCs are SSCs that are located near important SSCs such that changes to these SSCs could negatively impact the safety or mission of the facility or activity or function of the important SSCs. (DOE-STD-1073-2016, *Configuration Management*)
[5] Incorporates/supersedes VAR-10554 regarding optional use of P1040.

only software on that device. It can perform limited functions such as keypad controls or can provide more significant function and control capabilities.

2.3 Exclusions

This chapter does not apply to the following (Requirement 21-0103):

- Software covered by existing, compliant[6] software quality management plans/processes.
- Uncomplicated software tools, including productivity aids that are adequately controlled through the design process. Examples include unit conversion software, spike programs, equipment selection software, diagnostic software, and interpolation calculators
- Less Than Commercially Controlled (LTCC) software. Within Chapter 21, LTCC software is managed and controlled in accordance with P1040 (R16, §5.2.5).

  LTCC software does not require the completion of a Form 2033 or the development of a software quality management plan. (If a user is unsure if their software qualifies, they should fill out the Form 2033 to confirm.)

For **SSC Read-Only Embedded Software (aka Firmware)** only, Conditional, Partial Exclusions:

- ML-4 is excluded; however, the following are required:
  - Identify expected settings/set points in design documents.
  - Perform and document functional testing verifying the programming with reasonable assurance (PMT or Cx, ref. SOFT-V&V).
- ML-3 wherein LANL personnel (1) cannot review the software program nor (2) modify the firmware in any way (including entering/adjusting parameters/setpoints); however, perform the same two measures as ML-4 above, plus:
  - Complete a Form 2033 noting that "Software was not designed with LANL involvement and cannot be modified or configured, nor tested apart from the device in which it is embedded. SQA requirements are fulfilled by the QA requirements applied to testing, acceptance, and control of the entire device."
- ML-1 or ML-2: Same conditions for exclusion as ML-3 and same ML-3 compensatory measures, with these additional requirements:
  - The functional testing is to be done under a formal test plan addressing possible software faults, possibly including testing the device with invalid inputs, unexpected actions, and boundary conditions to ensure it works well in all situations, even unlikely ones (so-called negative testing).
  - When not from a qualified manufacturer (i.e., per IESL), CGD TEA plan shall consider means to increase confidence in manufacturer, possibly including assessment of software development ability and quality (ref. Section SOFT-ACQUIRE and AP-341-703, *Commercial Grade Dedication*).

3.0 CHAPTER OVERVIEW

Chapter 21 serves as the software quality plan for software following it. Implementation of this chapter will produce software-specific SQA documentation (e.g., software baseline, software data sheet, software approval for use) for specific software items. The software-specific documentation includes information unique to each software program, such as technical requirements and software ownership.

[6] Compliant means meeting the applicable SQA requirements of DOE O 414.1E and NQA-1 and includes compliance with P1040 or prior Chapter 21 revisions. When doubt exists, perform an assessment against the relevant requirements. Institutional Quality and Performance Assurance (IQPA) may be able to assist.

The chapter implements the SQA life cycle as shown in Figure 21.2. (Requirement 21-0104)

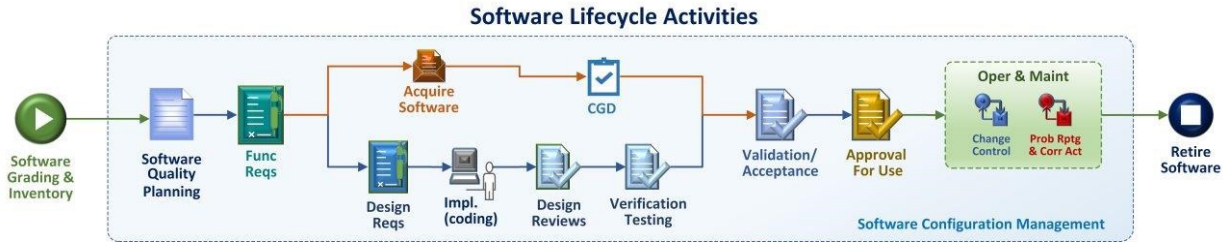


Figure 21.2 Software Life Cycle Activities

3.1 Chapter Organization

Chapter 21 is divided into six sections — SOFT-INTRO and five others that span the software life cycle: SOFT-GEN, SOFT-ACQUIRE, SOFT-DESIGN, SOFT-V&V, and SOFT-MAINT.

Note: Not all sections will apply to every piece of software, and section usage may not occur in sequential order.

- For all software, use SOFT-INTRO, SOFT-GEN, SOFT-V&V, and SOFT-MAINT.
- For software that is acquired, also use SOFT-ACQUIRE.
- For software that is designed and/or developed, also use SOFT-DESIGN.

Figure 21.3 shows an overview of the sections and of the deliverables that may be associated with each section. (Requirement 21-0105)

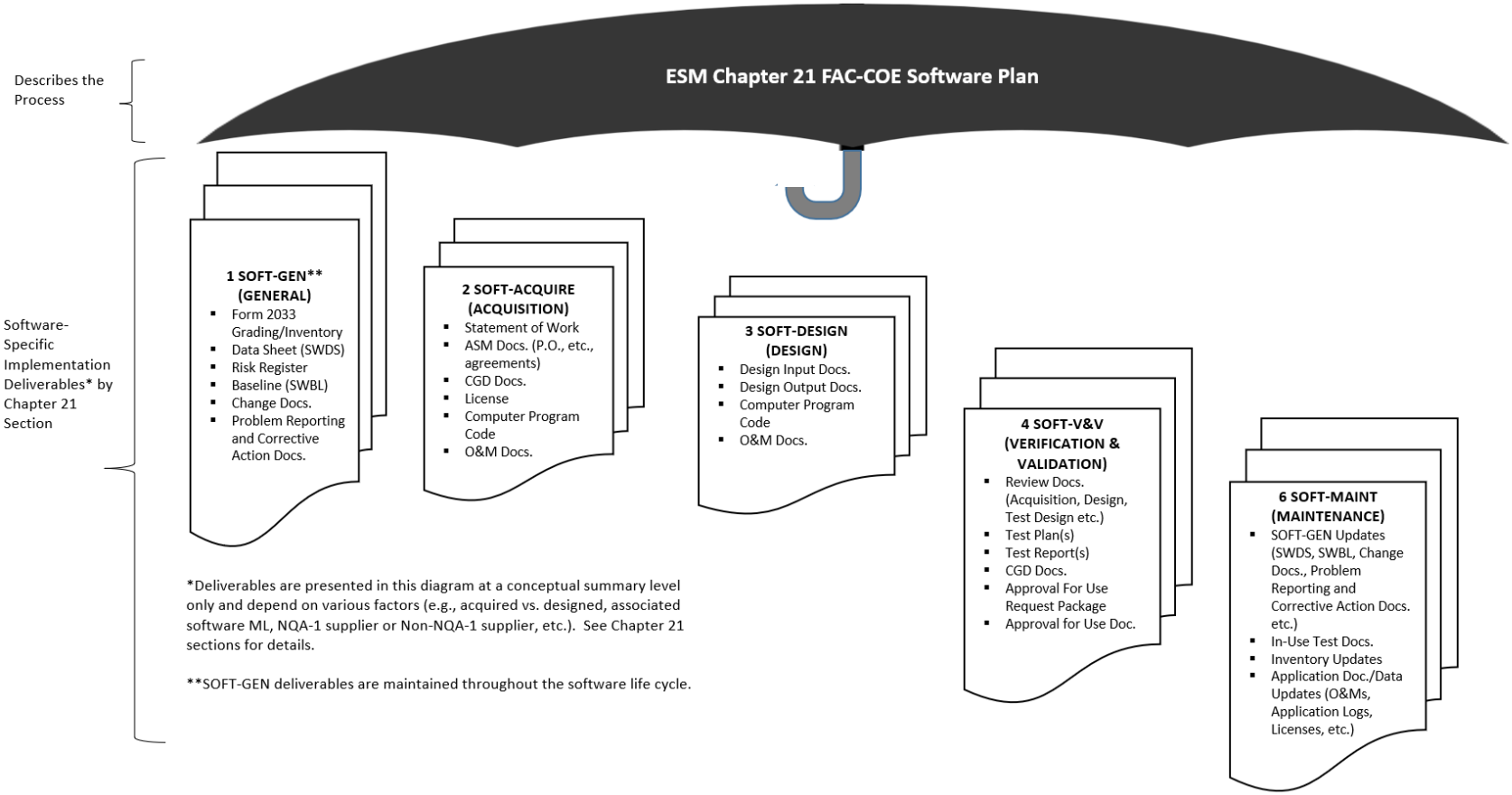


Figure 21.3 Chapter 21 Concept, Sections, and Deliverables

4.0 ROLES AND RESPONSIBILITIES

Roles and responsibilities are detailed in subsequent sections (e.g., by summary tables), but key roles include the Software Owner (SO) and Software Owner Responsible Line Manager (SRLM). The basic responsibilities of all SQA roles can be found in their definitions in SOFT-INTRO Appendix A *Definitions, Responsibilities, and Abbreviations*. Additional guidance can be found in SOFT-GEN Appendix A, *SO and SRLM Decision Diagram*.

5.0 GRADED APPROACH

Within this chapter, software Management Levels (MLs) are used to apply a graded approach to SQA requirements rather than software type and Software Risk Level (SRL), per P1040/Form 2033 software grading. Where ML-1, ML-2, ML-3, or ML-4 is used, this normally refers to the software ML.

- SSC software is assumed to have the same ML as the associated SSC unless an Equipment, Component, or Part Management Level Determination (MLDC) has been approved that states otherwise.
- Non-SSC software used for design or analysis may be used in activities involving SSCs with an ML equal to or lower than the software ML. Such software may only be used in design or analysis activities involving SSCs with a higher ML if the FDAR determines that use of the software can have no potential impact on the SSC's credited functions (e.g., through a component-level ML determination).

See COE AP-341-502, *Management Level Determination*, for details on ML and MLD.

As another example of grading or differentiation, SSC and non-SSC software may have differing requirements, as does acquired versus developed software. (Requirement 21-0106)

Figure 21.4 provides a simplified depiction of the graded approach utilized by Chapter 21 and of the software included within its scope, and outside of it (see Subsection 2.3 Exclusions for more on this).

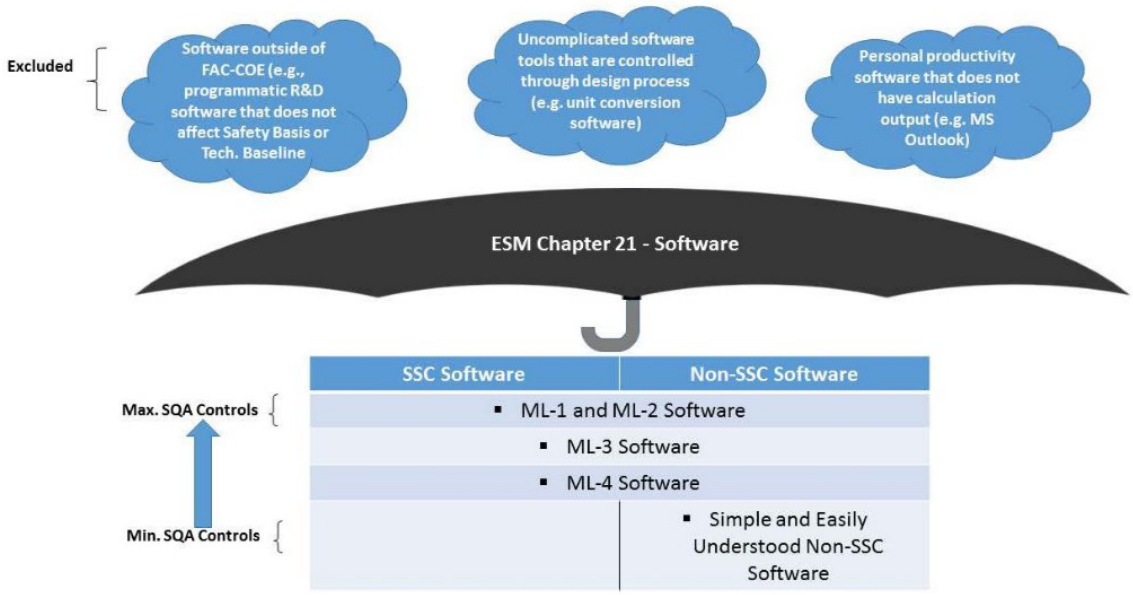


Figure 21.4 Chapter 21 Overview

6.0 IMPLEMENTATION

6.1 LANL

A. Once the requirements of this chapter have been adopted for a piece of software, compliance with the chapter must continue until software retirement.[7]

B. New software (SSC or Non-SSC) is required to comply with this chapter (subject to the exceptions and exclusions specified in subsection 2.3).

C. Before new equipment installed by existing projects may be operated, compliance with Chapter 21 is required[8] (subject to the exceptions and exclusions listed in subsection 2.3).

D. If existing software is identified as not managed and controlled by a compliant SQA program, the requirements of this chapter or P1040 shall be applied. In this scenario, initiate an issue (IM) per P322-4, *Issues Management*. The IM action must require the software to be compliant with either this chapter or P1040 by a date agreed to by both the SRLM and the chapter Point of Contact (POC).

Guidance: Existing software covered by a compliant software quality management program should periodically be evaluated to determine whether Chapter 21 should be adopted in place of the existing program. This evaluation should occur when:

– *A Major Computer Program Change (see definitions) is planned, or*

– *The management level (ML) of software increases.*

6.2 Subcontractors (Architect-Engineers, Consultants, Suppliers, Constructors, etc.)

A. For all SSC software (controls hardware real-time), provide a listing of such locations/usage to LANL (who, for ML-4, will satisfy required LANL software quality program expectations); ref SOFT-GEN/Software Summary (§*2.3*).

B. In addition, for SSC and Non-SSC ML-3 or higher software (e.g., nuclear or certain high-value, security, or environmental implications of failure beyond typical commercial), perform the following:

1. For subcontractors providing ML-1, ML-2, or ML-3 analysis or design deliverables, associated (non-SSC) software must be identified, categorized, and quality controlled at the appropriate stages in the design process (Ref. SOFT-GEN, etc.).

2. For SSC software (including reconfigurable firmware), subcontractor shall serve as the software owner (SO) up to the point of turnover. Upon turnover, transfer SO responsibility to LANL system engineer by way of the STR. (Note: The FDAR is the responsible manager throughout the software life cycle, however, may delegate Owner activities during development and implementation to the Design Agency or Constructor).

3. For software designed specifically for LANL, the supplier must perform V&V per Section SOFT-V&V as part of the supplier's software product development. When ML-1 or ML-2, suppliers must provide V&V processes and objective evidence of the supplier's V&V for LANL review and retention.

[7] Unless the SRLM or FDAR determines that use of P1040 is appropriate.
[8] This addresses projects underway that may not have included it in their scopes; requirement means a reasonable effort must be made to obtain and/or produce documents and comply with this chapter as if it had been part of the code of record since project initiation.

C. Beyond A–B above, further compliance with this chapter is only required when specifically invoked by subcontract[9].

Note to LANL: Flow down of additional Chapter 21 requirements to the subcontractor should be discussed with the chapter POC and defined in the Statement of Work (SOW). Generally, for higher MLs, additional requirements should be invoked unless, perhaps, the subcontractor has an existing SQA program that has been evaluated and determined to be compliant and/or LANL is willing to assume responsibility for ensuring that chapter requirements are met prior to project completion.

7.0 TRAINING AND QUALIFICATION

7.1 General

Training requirements are controlled by the Conduct of Engineering (CoE) office. At a minimum, Software Owners (SO) and Software Responsible Line Managers (SRLM) must read and be familiar with the current version of Chapter 21. In addition, it is recommended that SOs and SRLMs read and be familiar with the institutional document on SQA, P1040. It is important that the SRLMs and SOs understand what their responsibilities are for the software plans they manage. (Requirement 21-0107)

Note: *Training has been developed for this chapter. Both SOs and SRLMs should attend the overview (38047: CoE ESM Chapter 21 Software Overview, or equivalent), and SOs should also attend a more detailed course (34048: CoE ESM Chapter 21 Software Owners, or equivalent). Finally, the course "LANL Software Quality Assurance Overview Self-Study Course 60841" is generally informative, and has some P1040-specific content.*

7.2 Software Verification & Validation Testers

In Hazard Category 2 or 3 nuclear facilities, NQA-1 (Part I, Req. 2. 400) requires "certified" software testers for ML-1 and ML-2 software when final acceptance testing (which is nominally in the operating environment) and ideally all previous testing. Currently, the CoE Office does not maintain a policy for the training and qualification of such testers; therefore, each organization requiring such software testers shall qualify them and maintain documentation of same.

Form 2332 - Certification of Software Testers (CSWT) may be used to document qualification.

Note: *NQA-1 defines certification as the act of determining, verifying, and attesting in writing to the qualifications of personnel, processes, procedures, or items in accordance with specified requirements. For software testers, this means that qualification requirements have been established and that the qualifications have been met and "certified" (i.e., documented) by an appropriate manager.*

[9] This should involve a specific reference to Chapter 21 rather than a general statement about following the LANL ESM or Tailored Standards Manual.

APPENDIX A: DEFINITIONS, RESPONSIBILTIES, AND ABBREVIATIONS

Only key definitions are listed. Additional definitions and abbreviations are in GLOS-COE-1, the *Conduct of Engineering Glossary*, and those definitions supersede where they may conflict with this listing.

Table 21.A1 Definitions and Responsibilities

Item	Definition/Responsibility
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. NQA-1).
Acquired software	Software that is generally supplied through basic procurements, two-party agreements, or other contractual arrangements. Acquired software includes off-the-shelf (OTS) software such as operating systems, database management systems, compliers, 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 DOE G 414.1-4).
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).
Administrative controls	Administrative controls mean the provisions relating to organization and management, procedures, record keeping, assessment, and reporting necessary to ensure safe operation of a facility. (Ref. 10CFR 830)
Associated Management Level (ML)	For SSC software, the software ML is tied to its role in the SSC's function(s) — i.e., for the worst conceivable software failure, what SSC function(s) could be jeopardized, and what is the highest ML of those. For non-SSC software (used for design or analysis of an SSC), its ML will be the same as that associated with the SSC function it's being used to design or analyze. See AP-341-502 for more on ML determination.
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. NQA-1). ***Note:*** *See SOFT-GEN-FM02, Software Baseline Form (SWBL). 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.*
Bounding set points	Bounding set points are those that are bounding or limiting values required by or needed to satisfy safety basis requirements, protect equipment/systems from operational damage, or other limiting values for the proper intended operation of the software. Changes to operational set points within the minimum or maximum values would not constitute a change, but rather would be considered operational use of the software. Example: Bounding set points may be minimum or maximum pressure or tank level values.
Commercially Controlled (CC) software	As determined using Form 2033, Graded Software Inventory, lower risk software that does not meet the criteria of SS or [Risk Significant] RS software, but is higher risk than Less Than Commercially Controlled (LTCC) software. *[Chapter 21 uses MLs, not the CC/RS grading, so definition is only provided for information. Per P1040, CC category typically includes "commercial" (e.g., software updates, security patches), enterprise-level, and custom-developed workgroup software.]*

Table 21.A1 Definitions and Responsibilities	
Item	**Definition/Responsibility**
computer program	A combination of computer instructions and data definitions that enables computer hardware to perform computational or control functions. (Ref. NQA-1).
computer program listings	A printout [e.g., pdf] or other human readable display of the source and, sometimes, object statements that make up a computer program. (Ref. ISO/IEC/IEEE 24765).
configuration item	A collection of hardware or software elements treated as a unit for the purpose of configuration control. (Ref. NQA-1).
configuration management	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. NQA-1).
control point	A point in the software life cycle at which specified agreements or control (typically a test or review) are applied to the software configuration items being developed, e.g., an approved baseline or release of a specified document or computer program. (Ref. NQA-1).
design analyses	Calculations and/or experiments associated with design. (Based on NQA-1).
designed (or developed) software	Software that is designed or developed for a specific (custom) application. It may be developed by DOE or one of its Management and Operator contractors or contracted with a software company through the procurement process. ***Note:*** *Includes the following from DOE G 414.1-4. (a) custom-developed software, (b) configurable software, and (c) utility calculation software. (Definition developed for this chapter and based on DOE G 414.1-4).*
Designed Non- SSC software	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).
Designed SSC software	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).
embedded software	A more-preferred term for firmware, although read-only is most preferred.
Engineering Services Software Coordinator	Individual assigned to help maintain the non-SSC software inventory for ES Division, and to perform other duties as assigned (e.g., per desktop instruction).
error	An error is a condition deviating from an established baseline including deviations from the current approved computer program and its baseline requirements. (Ref. NQA-1). 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.
firmware	The combination of a hardware device, computer programs, and data that reside as read-only software on that device. The firmware (sometimes referred to as embedded software) can perform very limited functions such as keypad controls, or can provide significant function and control capabilities for control rod drives or safety systems. (Ref. NQA-1).
freeware	Software that is available for use at no cost or for a nominal, usually voluntary fee. (Ref. Merriam-Webster Dictionary).

Table 21.A1 Definitions and Responsibilities	
Item	**Definition/Responsibility**
function	A **function** is a task that must be performed. A function statement describes the capability necessary for a facility, system, or component to fulfill its mission. Describe a function using verb/noun combination such as "filter particulate." A function transforms inputs to desired outputs. In the example of "filter particulate," the function transforms an input fluid containing particulates into two outputs, the fluid without particulates and the particulates. A function describes what must be done, not how. Every function has at least one requirement associated with it.
hazard controls	Hazard controls mean measures to eliminate, limit, or mitigate hazards to workers, the public, or the environment, including: (1) Physical design, structural, and engineering features; (2) Safety structures, systems, and components; (3) Safety management programs; (4)Technical safety requirements; and (5) Other controls necessary to provide adequate protection from hazards. (Ref. 10CFR830).
Less-Than-Minor Computer Program Change	A change that is not a major or minor computer program change and: ▪ adds, deletes, and/or modifies ML-4 performance function code, ▪ adds, deletes, and/or modifies code that does not modify a Performance Function (all MLs), 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 soft start 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).
Major Computer Program Change	A change that: ▪ the Software Responsible Line Manager (SRLM) or computer program supplier designates as a Major Change, ▪ adds or deletes an ML-1, ML-2 or ML-3 SSC "Performance Function" (including bounding set point changes), ▪ modifies ML-1 or ML-2 SSC performance function code, excluding clarifying notes, ▪ adds, deletes or modifies design and/or analysis output values of ML-1, ML-2 or ML-3 calculations, ▪ recodes to another language, or ▪ modifies a significant number of lines of code. ***Note:*** *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.* Examples: 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).
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 P1020-2).

Table 21.A1 Definitions and Responsibilities	
Item	**Definition/Responsibility**
Minor Computer Program change	A change that is not a major computer program change and: ▪ adds or deletes an ML-4 SSC Performance Function (including bounding set point changes), ▪ modifies ML-3 SSC Performance Function code, excluding clarifying notes, or ▪ adds, deletes or modifies design and/or analysis output values of ML-4 calculations ***Note:*** 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. Examples: 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).
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 P1020-2).
negative testing	Testing the software with invalid inputs, unexpected actions, and boundary conditions to ensure it works well in all situations.
Non-NQA-1 qualified supplier	A supplier that did not develop and maintain the software in accordance with an NQA-1 quality assurance program. (Definition developed for this chapter).
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, *Guidance on the Development, Evaluation, and Application of Environmental Models*.)
Non-safety software	As determined using Form 2033, software that is not determined to be safety software. Non-safety software includes risk significant, commercially controlled, and less than commercially controlled software. (Ref. P1040).
Non-SSC software	Software used in design, analysis and/or for administrative control. This software does not physically monitor and/or control SSCs. Examples: 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).
Otherwise-Acquired Software	Software that was not acquired, developed and/or maintained in accordance with an NQA-1 quality assurance program. This software may be from entities internal to LANL entities external to LANL (e.g., other DOE sites, U.S. EPA, etc.) This includes existing software (also referred to as in-use or legacy software).
operating environment	A collection of software, firmware, and hardware elements that provide for the execution of computer programs. (Ref. NQA-1). It is also the location and conditions (environment) where the software will be used or operated to meet its intended function. (Based on P330-8).
performance function	A function that is required to satisfy item performance. (Definition developed for this chapter).
performance function code	The computer program language (code) that is required to satisfy item performance. (Definition developed for this chapter). The performance function code is only those lines of code that affect the performance function.

Table 21.A1 Definitions and Responsibilities	
Item	**Definition/Responsibility**
positive testing	Testing the software using expected inputs.
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 sometimes used.) **Note:** 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).
Regression testing	Selective retesting to detect errors introduced during modification of the computer program or to verify that the modified computer program still meets its specified requirements. (Ref. NQA-1).
Risk Significant (RS) software	As determined using Form 2033, software that is similar to [safety] software except that it is used for work that is not nuclear or radiological, but is potentially hazardous to LANL workers and the environment (e.g., accelerator, live-firing range, biological hazard facility, high explosive facility, or moderate- or high-chemical hazard facility). RS software also includes software that is used in support of a LANL Mission Essential Function (MEF). *[Note: RS is only relevant when using P1040 rather than Chapter 21 for SQA, since Chapter 21 uses MLs, not RS (or Commercially Controlled) for grading/actions]*
Safety software	Includes any of the following: • performs a safety function as part of a Facility Structure, System, or Component (SSC) and is cited in either a DOE-approved documented safety analysis (DSA) or an approved hazard analysis, • classifies, designs, or analyzes nuclear facilities to ensure the proper accident or hazards analysis of nuclear facilities, • classifies, designs, or analyzes an SSC that performs a safety function, • performs a hazard control function in support of nuclear facility, radiological safety management program, or technical safety requirements (TSRs), or • performs a control function necessary to provide adequate protection from nuclear facility or radiological hazards and supports eliminating, limiting, or mitigating nuclear hazards to workers, the public, or the environment. (Ref. P1040, R16).
Simple and Easily Understood (Non-SSC) software	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 is individually verified with each use (e.g., calculation). (Based on NQA-1).
software	Computer programs and associated documentation and data pertaining to [needed for] the operation of a computer system. (Ref. NQA-1).
software approval/ approved for use (SWAU)	The process of ensuring and documenting that the software requirements have been satisfied (including installation and operating instructions) and the software is ready to be used in its intended operating environment. (Ref. P1040).
software change	A software change is an addition, deletion and/or modification to software. (Definition developed for this chapter).

Table 21.A1 Definitions and Responsibilities	
Item	**Definition/Responsibility**
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). *ES-Div has historically had such, who helps with non-SSC, Facility Engineering software inventory per SOFT-MAINT.* ES-Software@lanl.gov
software design requirement	A requirement that impacts or constrains the design of a software system or software system component. (Based on ISO/IEC/IEEE 24765).
software engineering	(a) the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software (b) the study of approaches as in (a) (Ref. NQA-1)
software life cycle	The period of time that begins when a software product is conceived and ends when the software is no longer available for use. The life cycle typically includes a concept phase, requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and, sometimes, retirement phase. These phases may overlap or be performed iteratively, depending on the software development approach used. (Ref. NQA-1)
software design verification:	The process of determining if the product of the software design activity fulfills the software design requirements. (Ref. NQA-1).
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. NQA-1).
Software Owner (SO)	Responsibilities (performs or causes to be performed; see others in chapter): • Provides the software information and Form 2033 and obtains review and concurrence of the form in accordance with this document. • Reviews and approves the software project planning documentation. • Owns the software and supports the SRLM in complying with the requirements of this document. • Prepares the approval-for-use documentation that describes the intended use and any associated limitations, access controls, etc., for using the software.
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 is the main or super-user of each program but can be another user or even a single individual for a group or division.
Software Repository	A shared storage location that is defined by a written procedure detailing the location, access and other control methods, and formats used to store the software.

Table 21.A1 Definitions and Responsibilities	
Item	**Definition/Responsibility**
Software Responsible Line Manager (SLRM)	Responsibilities (performs or causes action below to be performed; see others in chapter): • Manages and maintains software in accordance with this document to ensure it operates as intended. • Reviews and signs Form 2033 as a record in the Graded Software Inventory (GSI). • 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. • Provides software inventory information and any changes thereto to QPA-IQ and ES. • Applies the appropriate amount of SQA rigor in software planning and implementation. • Reviews and approves the software project planning documentation. As applicable, acquires software and/or software services. • Except for SQA associated with using the software, is responsible for the SQA of the software. • V&Vs the software. • 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). • Ensures software owning organization personnel managing or working to this document are adequately trained and as required, qualified. • Approves software for use. • Completes in-use tests in the operating environment. • Retires software, including removal of safety software from software inventories.
software tool	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 NQA-1).
Software User (SU)	Responsibilities (performs or causes to be performed, see others in chapter): • Reports software errors and problems. Uses software within software limitations and in accordance with this document.
Software User Responsible Line Manager (SU RLM)	Responsibilities (performs or causes actions below to be performed, see others in chapter): • Supports completion of in-use tests in the operating environment. Ensures software users and software user organization personnel managing or working to this document are adequately trained, and as required, qualified.
SSC software	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. Examples: 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, *Quality Assurance for Software Important to Safety*, IAEA, 2000).

Table 21.A1 Definitions and Responsibilities	
Item	**Definition/Responsibility**
support software	Software or a program that aides in the development, maintenance, or use of other software or provides general application-independent capability (Ref. ISO/IEC/IEEE 24765). Support software includes software tools and system software (Ref. NQA-1). **Note***: SSC and Non-SSC software may have support software.*
system software	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 NQA-1).
test case	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. NQA-1).
testing (software)	The process of: (a) operating a system (i.e., software and hardware) or system component under specified conditions (b) observing and recording the results (c) making an evaluation of some aspect of the system (i.e., software and hardware) or system component in order to verify that it satisfies specified requirements and to identify errors (Ref. NQA-1)
test plan (procedure)	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. NQA-1).
toolbox code	Software that is listed on the DOE Safety Software Quality Assurance Central Registry (Registry). *This is historically only for Safety Basis software.*
validation (software)	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. NQA-1); 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. DOE O 414.1E).
verification	The act of reviewing, inspecting, testing, checking, auditing, or otherwise determining and documenting whether items, processes, services, or documents conform to specified requirements (Ref. NQA-1); 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. DOE O 414.1E).

Table 21.A2 Abbreviations	
Term[1]	**Definition**
A	Approve
ASCE	American Society of Civil Engineers
AP	Administrative Procedure
ASM	Acquisition Services Management
ASME	American Society of Mechanical Engineers
BAS	Building Automation System
CAM	Continuous Air Monitor
CGD	Commercial Grade Dedication (Dedicated)
CM	Configuration Management
CoE	Conduct of Engineering
COOP	Continuity of Operations Plan
COTS	Commercial Off-the-Shelf
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	Guide
Gr	Grade
GSI	Graded Software Inventory
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	ASME NQA-1-2008/NQA-1A-2009, Quality Assurance Requirements for Nuclear Facility Applications, Part I and Part II

Table 21.A2 Abbreviations	
Term[1]	**Definition**
OCIO	Office of Chief Information Officer
O&M	Operations and Maintenance
OTS	Off the Shelf
PFD	Process Flow Diagram
P&ID	Process and Instrumentation Diagram
POC	Point of Contact
PR&CA	Problem Reporting and Corrective Action
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
SCM	Software Configuration Management
SDD	System Design Description
SME	Subject Matter Expert
SO	Software Owner
SOO	Sequence of Operations
SOW	Statement of Work
SPOC	Software Point of Contact
SQA	Software Quality Assurance
SRLM	Software Responsible Line Manager
SSC	Structure, System, or Component
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
SWRR	Software Risk Register
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)

[1] **Only key terms are listed. See GLOS-COE-1 for additional ones.**

APPENDIX B: CHAPTER 21 REFERENCE LISTING

The most recent document revision (including revised document numbering and/or titles) applies unless otherwise stated. (Requirement 21-0108) *Most national standards are available to LANL personnel via links; others must purchase.*

Non-LANL (Publicly Available)

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*

ASQ *Certified Software Quality Engineer Handbook*, L. Westfall, Am. Soc of Quality Pres.

DOE G 413.3-21, *Cost Estimating Guide*

DOE O 414.1E, *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 G 450.4-1C, *Integrated Safety Management System Guide*

DOE P 450.4A, *Integrated Safety Management Policy*

DOE SQAS21.01.00-1999, *Software Risk Management–A Practical Guide*

DOE-STD-1073, *Configuration Management*

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*

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 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 29119, *Software and systems engineering—Software testing*

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*

Merriam-Webster Dictionary

LANL (internal-only links except as noted)

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*

AP-341-502, *Management Level Determination*

AP-341-504, *Temporary Modification Control*

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 3400.00.0410, *Goods or Services Requiring Special Review/Approval*

ASM Website

Conduct of Operations Resources

CIO-P100, *Cybersecurity Program Plan*

SEO-COOP-006, *Continuity of Operations Plan*

ES Division Office (ES-Div) Website

ESDO-AP-002, *Engineering Services (ES) Software Inventory Instructions*

Form 2033, *Graded Software Determination*

Forms for P1040, many of use herein

Information System Security Officer (ISSO)

Institutional Evaluated Supplier List (IESL)

Institutional Quality SQA webpage

IQPA-IQ-FSD-116, Software Quality Management Plan (SQMP) for LANL Less Than Commercially Controlled (LTCC) Software

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

LIST-SB-503, *Nuclear Facility List,* (e.g., r2.1 directly here, or under Other Resources)

P204-1, *Controlled Unclassified Information*

P218, *Cyber Security Controls*

P322-4, *Issues Management*

P328-5, *Assessments*

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

P330-5, *Special Processes*

P330-6, *Nonconformance Control and Reporting*

P330-8, *Inspection and Test*

P330-9, *Suspect/Counterfeit Items (S/CI)*

P341, *Facility Engineering Processes Manual*

P342, *Engineering Standards*

P343, *Facility Engineering Training and Qualification Manual*

P781-1, *Conduct of Training*

P805, *Export Control*

P841-1, *Quality Procurements*

P1020-2, *Laboratory Document Control*

P1040, *Software Quality Management*

P1011, *Information Technology Acquisition and Use*

PD110, *Safety Basis*

PD115, *LANL Readiness Program*

PD210, *Cyber Security Program*

PD330, *LANL Quality Assurance Program Description*

PD340, *Conduct of Engineering and Configuration Management for Facility Work*

PD370, *Conduct of Engineering for Research and Development (R&D)*

Safety Basis Document List

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*

SD130, *Nuclear Criticality Safety Program*

Software Quality Assurance webpage (IQPA-IQP)

STD-342-100, *Engineering Standards Manual* (available externally)

TeamForge

Contents

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, *Software Risk Register*; DOE-STD-1073, *Configuration Management*. 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.

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 summary[1] 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.

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 it 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 before or after 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 before 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 after 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 ML[1]				Non-SSC Software ML[1]				
		1	2	3	4	1	2	3	4	
[1] See Form 2033 for software type and software risk levels. R = Required. Gr = Required but graded. "-" = Not required. [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. [3] Use SOFT-GEN-FM02, *Software Baseline Form (SWBL)*. Include documents and computer program files. [4] See SOFT-V&V; ESM Chapter 15, *Commissioning;* and AP-341-801, *Post Modification/Post Maintenance Testing* for related testing requirements of SSC software changes.										

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

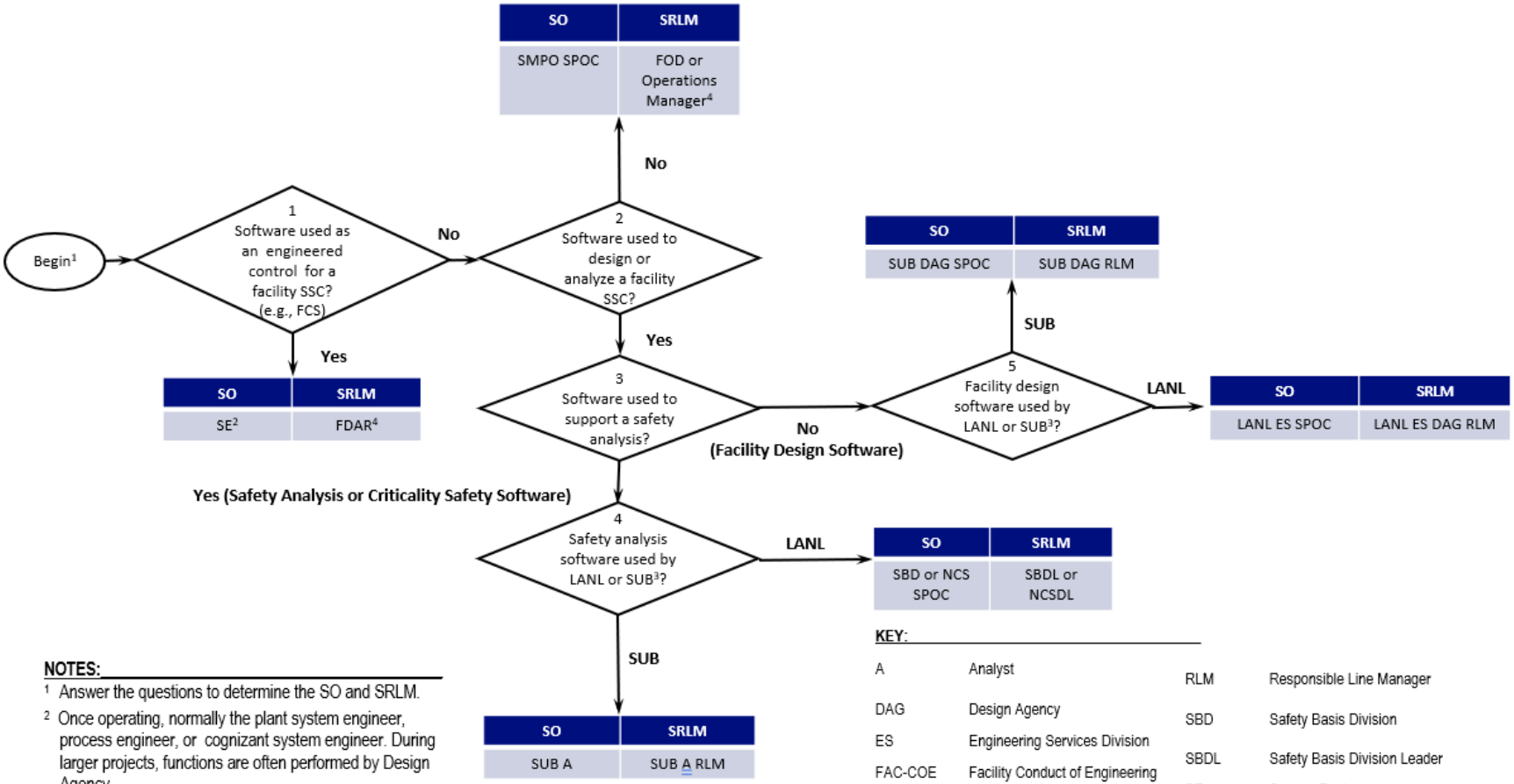
APPENDIX A: CHAPTER 21 SO AND SRLM DECISION DIAGRAM (GUIDANCE)

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.

KEY:

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

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 before or after software approval for use (SWAU)	Documented comment in the computer program where feasible	R	R	R	R	R	R
	Minor or major changes made before 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 after 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: ○ SSC drawings ○ SSC performance specification ○ FDD or SDD ▪ Other SSC design inputs/outputs as required	R	Gr	Gr	-	-	-
		▪ 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)	▪ Software design outputs: ○ 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 terms[10]	R	Gr	Gr	R	Gr	-[13]
SOFT-DESIGN, 5	Translate the software design into computer program language (code)	▪ Computer program language (code): ○ 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	▪ Dispositioned review comments	R	Gr	Gr	R	Gr	Gr
SOFT-V&V, 5.1	Review acquisition	▪ Acquisition review document	-	-	-	R	R	-
SOFT-V&V, 5.2	Develop and review the test plan (SWTP)	▪ 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)	▪ Software design ▪ Dispositioned review comments	R	Gr	Gr	R	Gr	Gr
SOFT-V&V, 4.4	Review computer program code where feasible	▪ 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	▪ 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)	▪ 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	▪ 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	▪ 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.

Contents

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, *Software Risk Register*; DOE-STD-1073, *Configuration Management.* 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	Merged/streamlined SSC and Non-SSC requirements while maintaining essential content.	Tobin Oruch, *ES-FE*	Michael Richardson, *ES-DO*

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

SOFT-ACQUIRE: Acquiring Software

1.0 INTRODUCTION

1.1 Purpose

This section applies to workers who acquire software. (See SOFT-INTRO for definition of acquired). Software acquisition includes both software and software services, whether procured (purchased) or acquired in other ways.

Before using this section, perform the applicable activities described within SOFT-GEN (e.g., software identification, completion of a draft or final Software Grade Determination (Form 2033), and other steps).

1.2 Definitions and Abbreviations

Appendix A of SOFT-INTRO addresses defined Chapter 21 terms.

1.3 Summary Table

The following table provides an overview of the requirements and deliverables of this Section. **It also addresses who performs tasks, and when.**

Table 21.3 SOFT-ACQUIRE Section Summary for Structure, System, and Component (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[1]**			**Implementation Detail**			**Reference**
		1, 2	3	4	**How**	**When**	**Who[2, 3]**	**ESM Ch. 21 Ref.[8]**
I. For SSC Software								
1	Acquire new software	R	Gr	Gr	▪ Evaluate and document whether to acquire from a Nuclear Quality Assurance (NQA)-1 or Non-NQA-1 qualified supplier and commercially dedicate (CGD)[4] ▪ Develop a software data sheet (SWDS), (SOFT-GEN-FM01)[5] ▪ Develop Statement of Work (SOW)[6] ▪ Acquire per P841-1, PD210 and CIO-P100 ▪ Develop interface agreement, integrated contract orders, interagency agreements or equivalent[7] ▪ Complete Exhibit H[5]	▪ When needed to perform work	▪ SO (D) ▪ SRLM (R, A)	SOFT-GEN 2.0 (all) SOFT-GEN 3.0 (all) SOFT-ACQUIRE, 2.0 (all)
II. For Non-SSC Software								
1	Acquire existing LANL software	R	R	Gr	▪ Look for and evaluate existing LANL software for usage ▪ If suitable, acquire existing software from the SRLM and obtain SRLM authorization for use ▪ Update documentation (e.g. licenses), as applicable, to reflect additional authorized user(s)	▪ Look for and evaluate existing LANL software before acquiring new software ▪ When needed to perform work	▪ SU (D) ▪ SRLM (R, A)	SOFT-ACQUIRE (all)

Table 21.3 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[1]			Implementation Detail			Reference
		1, 2	3	4	How	When	Who[2, 3]	ESM Ch. 21 Ref.[8]
2	Acquire new software	R	Gr	Gr	▪ Determine and document if software is used in SSC design and is *simple and easily understood*; document and verify with each use ▪ Determine if software is a toolbox code ▪ Evaluate and document whether to acquire from NQA-1 or Non-NQA-1 qualified supplier and commercially dedicate the software (CGD)[4] ▪ Develop SWDS, (SOFT-GEN-FM01)[5] ▪ Develop SOW[6] ▪ Acquire per P841-1, PD210 and CIO-P100 ▪ Develop CGD documentation per AP-341-703[4] ▪ Use P-card or purchase order (PO) ▪ Develop inter- or intra-LANL agreements[7] ▪ Complete Exhibit H[5]	▪ When needed to perform work	▪ SO (D) ▪ SRLM (R, A)	SOFT-ACQUIRE (all) SOFT-GEN 3.0 (all)

Table Notes:
[1] ML = Associated Management Level per Form 2033. Complete Form 2033 prior to acquisition. R = Required. Gr = Required but graded.
[2] D = Develop or implement; R = Review; A = Approve.
[3] SO = Software Owner. SRLM = Software Responsible Line Manager. SU = Software User
[4] CGD applies to ML-1 and ML-2 software only.
[5] Required for ML-1 through ML-3 software or for any ML if the supplier is required to come onsite to install/maintain the software.
[6] Applies to ML-1 through ML-3 software only.
[7] See P841-1 for details. Required for acquisitions from LANL organizations, DOE sites, and/or government agencies.
[8] Ref. = ESM Chapter 21 section name and subsection number (e.g., Section SOFT-GEN, Subsection 3, Software Identification and Determination).

2.0 SSC AND NON-SSC SOFTWARE ACQUISITION

2.1 General Requirements

The requirements in this subsection apply to the acquisition of all software managed under Chapter 21. (Requirement 21-0301)

NOTE: Before acquiring new Non-SSC software, look for and evaluate whether existing LANL software may be available to use. If existing software is suitable, then acquire using methods described in this chapter. For Facility Engineering divisions, inventory is per SOFT-MAINT (§4.1).

A. The software acquisition process must ensure that malicious software, including suspect/counterfeit, is not permitted to enter the LANL supply chain or computing environment.

B. Acquire software in accordance with P841-1, Quality Procurements, PD210, Cyber Security Program, CIO-P100, Cybersecurity Program Plan, CIO-P215, Industrial Control Systems (ICS) Risk Management, and the following:

1. Ensure software functional requirements are fully identified, documented, reviewed and approved.

Note: It may not be necessary to develop specific requirements for firmware if the documented requirements for the SSC define the functional requirements for the associated firmware.

2. Ensure the software data sheet (SWDS) is complete, if required.

a. As required, clarify LANL-supplier division of responsibility on the SWDS.

Example: LANL plans to acquire software for a management level (ML)-2 application. The supplier only provides notifications of defects, new releases, or other issues that may impact the software operation on the supplier's website. The SWDS specifies that, prior to each use, the software responsible line manager (SRLM) must review the supplier website and/or contact the supplier to ensure this problem reporting (PR&CA) notification/action requirement is fully satisfied.

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

3. Ensure the software identification number (SWID) is included in the acquisition documentation.

4. Ensure all applicable requirements (including those from subsequent subsections) are documented in the Exhibit H (or Statement of Work) associated with the procurement.

5. Engage the LANL chief information officer (CIO) as necessary to satisfy the requirements of CIO-P100, CIO-P215, and other applicable procedures.

6. When subcontractors are involved in the development and/or supply of software to LANL, ensure that all applicable software quality requirements are included in the associated statement of work. Generally, the following requirements should be flowed down:

- For SSCs with software being specified or designed, and for any software being delivered for LANL ownership and use, require the same documentation deliverables that LANL personnel must provide (to the extent

the information is known in the design and construction phases; when not known, insert TBDs and deliver as “Draft”).

- Unless otherwise indicated in subcontract documents, the construction subcontractor serves as the software owner (SO) up to the point of turnover. Upon turnover, the SO responsibility transfers to the LANL system engineer or other designated SO. The facility design authority representative (FDAR) is the SRLM and retains that responsibility until turnover, however, they may delegate activities through subcontract to the subcontractor.
- Subcontractor quality assurance programs must meet the requirements of this chapter, comply with DOE O 414.1E, and (where required by subcontract) ASME nuclear quality assurance (NQA)-1, including Part I and Part II, Subparts 2.7 and 2.14.

2.2 SSC and Non-SSC Software (ML-1 through ML-3)

The following requirements shall be addressed in procurement documents for all ML-1 through ML-3 software except as noted.

A. From NQA-1 qualified suppliers (per IESL) for ML-1 and -2, and when available and desired 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. (Requirement 21-0302)

B. ML-1 and ML-2 only: For software that will undergo CGD, specify any supplier support required to dedicate the software (e.g., access for supplier assessments, surveys, third-party dedicators). (Requirement 21-0303)

The dedication process shall be documented and include the following:

- Identification of the capabilities and limitations for intended use as critical characteristics; and,
- Use of test plans and test cases as the method of acceptance to demonstrate the capabilities within the limitations; and,
- Instructions for use (e.g., user manual) within the limits of the dedicated capabilities.

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.

For guidance in dedicating computer programs, see AP-341-703 and ASME NQA-1-2015 Part III, Subpart 3.2-214, Quality Assurance Requirements for Commercial Grade Items and Services, Commercial Grade Computer Programs, and Software Services and the references therein.

Subsequent revisions of ML-1 and ML-2 software from non-NQA-1 qualified suppliers that do not follow NQA-1 Subpart 2.7 to V&V the software for acceptance (as described in SOFT-V&V) may need to be dedicated. See Section 5.2.C of SOFT-GEN for details.

Incorporate the following guidance as applicable, when dedicating computer programs (Ref. NQA-1-2015, Part III, Subpart 3.2-2.14):

Guidance: 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.

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.

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.

C. Specify any required documentation (and/or computer programs) of methods used in the development and validation of the software (e.g., test cases).

D. Specify any requirements for supplier notification of new releases or other issues (in addition to errors) that could potentially impact the software operation.

E. Specify any training and/or technical support required for successful installation and/or use of the software.

F. The following requirements shall be addressed in procurement documents for ML-1 through ML-3 non-SSC software.

- Specify any required design codes for the software (e.g., SAP2000 must satisfy ASCE 10-15 for steel frame design and ACI 318-19(22) for concrete frame design).

CONTENTS

RECORD OF REVISIONS

Rev.	Date	Description	POC	RM
0	06/23/2016	Initial issue	Tobin Oruch, *ES-DO*	Mel Burnett, *CENG-OFF*
1	05/25/2017	Added design output clarifications and other minor editorial clarifications	Tobin Oruch, *ES-DO*	Lawrence Goen, *ES-DO*
2	02/11/2026	Streamlined while maintaining most essential content, and merged SSC and Non-SSC requirements	Tobin Oruch, *ES-FE*	Michael Richardson, *ES-DO*

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

SOFT-DESIGN: SOFTWARE DESIGN

1.0 INTRODUCTION

1.1 Purpose

SOFT-DESIGN describes Chapter 21 requirements for designing (developing) software.

1.2 Applicability

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

1.3 Resources

Document examples and templates relevant to software design activities are provided on the Chapter 21 website under SOFT-DESIGN. They are for informational purposes only, may be used as a guide when creating documentation.

The documents may include the following:

- Form 2323, Software Requirements Specification (SRS) (docx)
- Software requirements specification (SRS) example (Source Tracker)
- Design document template and example for designed software (Source Tracker)
- User manual (O&M instructions) for designed software (Source Tracker)

Additional deliverables will generally be required.

1.4 Summary Tables

The following tables provide an overview of the requirements and deliverables of this Section. **They also contain details on who performs related tasks, and when.**

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[1]		Implementation Detail			Reference
		1, 2	3, 4	How	When	Who[2, 3]	ESM Ch. 21 Ref. Section[4]
1	Develop software design requirements (design input)	R	Gr	▪ Develop software design input deliverables: ○ SSC drawings (e.g., PFDs, P&IDs, SOOs, hardware/instrument location drawing, location drawing, instrument/device list) ○ SSC performance specification ○ FDD or SDD ○ Other SSC design inputs/outputs as required	▪ Earliest practical time ▪ 60%, 90%, 100% SSC design ▪ Prior to software design as much as practical	▪ SSC DA ▪ (D, R, A) LANL SRLM (R, A) ▪ FDAR (R, A)	SOFT-DESIGN, 3
2	Design the software (design output)	R	Gr	▪ Design the software (output) deliverables: ○ Software specification (SWRS) ○ Software design (SWDD) ○ Software system hazard analysis and mitigation documentation (SWHA; optional) ○ Software design traceability documentation (SWTM) ○ Operations and maintenance (O&M) instructions ○ SOFT-GEN deliverables ○ Other software design deliverables as required	▪ Earliest practical time ▪ If in scope of SSC design, software requirements (draft) at 30% design submittals, firmed up at 60% ▪ If not in scope of SSC design, then in detail design submittals ▪ Before software coding as much as practical	▪ SD (D, R, A) ▪ SRLM (R, A) ▪ LANL SRLM (R,A)	SOFT-DESIGN, 4
3	Translate the software design into computer program language	R	Gr	▪ Develop computer program language (code): ○ Data structures/files ○ Source code (where feasible) ○ Executable code ▪ Provide computer program listings (e.g., PDF printout of source code)	▪ Earliest practical time ▪ Before final acceptance testing	▪ SD (D, R, A) ▪ SRLM (R, A)	SOFT-DESIGN, 5

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[1]		Implementation Detail			Reference
		1, 2	3, 4	How	When	Who[2, 3]	ESM Ch. 21 Ref. Section[4]

[1] ML = associated management level per Form 2033; R = required; Gr = required but graded

[2] D = develop or implement; R = review; A = approve

[3] SSC DA = design agency for SSC that is associated with the software; SRLM = software responsible line manager (before turnover to LANL); LANL SRLM = SRLM after turnover to LANL; FDAR = facility design authority representative

[4] Ref. = ESM Chapter 21 section name and subsection number (e.g., Section SOFT-GEN, Subsection 3, *Software Identification and Determination*)

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[1]		Implementation Detail			Reference
		-1, -2	-3, -4	How	When	Who[2, 3]	ESM Ch. 21 Ref.[5]
1	Develop software design requirements (input)	R	Gr	▪ Develop system requirements (input) specification (SWYRS)	▪ Earliest practical time ▪ Before software design as much as practical	▪ SO (D, R, A) ▪ SRLM (R, A) ▪ FDAR (R, A)	SOFT-DESIGN, 3
2	Design the software (output)	R	Gr	▪ Design the software (output) deliverables: ○ Software specification (SWRS) ○ Software design (SWDD) ○ Software system hazard analysis and mitigation documentation (SWHA; optional) ○ Software design traceability documentation (SWTM) ○ O&M instructions ○ SOFT-GEN deliverables ○ Other software design deliverables as required ○ Computer model mathematical terms[4]	▪ Earliest practical time ▪ Before software coding as much as practical	▪ SD (D, R, A) ▪ SRLM (R, A)	SOFT-DESIGN, 4

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[1]		Implementation Detail			Reference
		-1, -2	-3, -4	How	When	Who[2, 3]	ESM Ch. 21 Ref.[5]
3	Translate the software design into computer program language (code)	R	Gr	▪ Develop computer program language (code): ○ Data structures/files ○ Source code (where feasible) ○ Executable code ▪ Provide computer program listings (e.g., PDF printout of source code)	▪ Earliest practical time ▪ Before final acceptance testing	▪ SD (D, R, A) ▪ SRLM (R, A)	SOFT-DESIGN, 5

[1] ML = associated management level per Form 2033; R = required; Gr = required but graded
[2] D = develop or implement; R = review; A = approve
[3] SO = software owner; SRLM = software responsible line manager; FDAR = facility design authority representative
[4] Applicable to computer program models only.
[5] Ref. = ESM Chapter 21 section name and subsection number (e.g., SOFT-GEN, Subsection 3, *Software Identification and Determination*)

2.0 DEFINITIONS AND ABBREVIATIONS

Appendix A of SOFT-INTRO addresses defined Chapter 21 terms.

3.0 SOFTWARE DESIGN INPUTS

This subsection describes the minimum requirements, activities, and deliverables for preparing software design requirements (i.e., inputs). Software design inputs are provided to the software designer. (Requirement 21-0401)

A. Software design requirement (input) deliverables depend on the nature, complexity, and associated ML of the software. Common software design input deliverables include the following:

- Software requirements (e.g., performance specification, software requirements specification [SRS], or requirement definition in the statement of work [SOW])
- System drawings (e.g., process flow diagrams [PFDs], process and instrumentation diagrams [P&IDs], sequences of operations [SOOs], hardware/instrument location drawing, network drawing[1], instrument/device list)
- Facility design description (FDD), system design description (SDD), or both
- Other design inputs as required to support software design[2]

B. Software design requirements (inputs) must

1. be identified and documented using a systems engineering process (Ref. P341, *Facility Engineering Processes Manual,* and ESM Chapter 20);
2. be based on upper-tier performance and functional requirements.
3. identify the operating system, function, interfaces, performance requirements, installation considerations, applicable SSC (structure, system, and component) design inputs, critical characteristics (as applicable), and design constraints of the computer program (Requirement 21-0402);
4. specify technical and software engineering requirements, including security features;
5. identify applicable reference drawings, specifications, codes, standards, regulations, procedures, or instructions that establish testing, inspection, and acceptance criteria;
6. be commensurate with the risk of unauthorized use and address security requirements (e.g., vulnerability/cyber-security protections; see SOFT-ACQUIRE §2.1); (Requirement 21-0403)
7. be traceable throughout the software life cycle (ML-1 through ML-3 software);
8. consider the computer program's operating environment (see definitions in SOFT-INTRO);
9. identify and address any user human-machine interface (HMI) requirements, factor in existing user operational protocols and the conventions and methods

[1] Level of detail as required by facility design authority representative (e.g., a notional or "block" diagram of the network at a minimum, which will later become a detailed technical drawing)
[2] As determined by the SO, SRLM, or both; may include hazard analyses, requirements criteria document (RCD), identification of commercial grade dedication (CGD) critical characteristics, etc.

employed by users/operators, and as appropriate, specify prototypes/HMI screenshots for review and comment as part of the design deliverables; and

10. include user-level input/review of screenshots, prototypes, etc., as applicable.

For guidance on software design inputs, 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.

Table 21.3-3 Software Design Input Deliverables

No.	Deliverable
01	Software requirements (e.g., performance specification, software requirements specification, or listing in SOW)
02	For SSC: system drawings (e.g., PFDs, P&IDs, SOOs, hardware/instrument location drawing, network drawing[1], instrument/device list)
03	For SSC: FDD, SDD, or both
04	Other design inputs/outputs as required to support software design[2]

Note: Required for all software ML-1 through -4 or other ESM chapters.
1. Level of detail as required by FDAR (e.g., a notional or "block" diagram of the network at a minimum, which will later become a detailed technical drawing)
2. As determined by SRLM; may include hazard analyses, RCD, identification of CGD critical characteristics, etc.

C. Additionally, SSC software design requirements (inputs) must

1. be documented as part of the SSC design documents (e.g., P&IDs, SOOs, system specifications, SDDs);
2. be based on system/component hazard analyses that identify system/component risks and the means for controlling them (for ML-1 through ML-3 software); and
3. be 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*.

D. Ensure the design inputs and sources are identified and documented, specified on a timely basis, and translated into design documents — and that their selection is reviewed and approved.

E. Finally, these requirements must call for the processing and retention of software design input documentation in accordance with the SRLM's governing document control and records management process. It is important to ensure that the software identification number is part of the documentation record number.

4.0 SOFTWARE DESIGN OUTPUTS

Design software at the earliest practical time and complete as much of the design as practical before 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 the software design deliverables as part of the required submittal process.

Deliverables and documentation must be reviewed during the software design process and are subject to software design verification requirements in accordance with Section 4.0 of SOFT-V&V.

4.1 General Software Design Requirements

This Subsection provides a high-level description of the software design output requirements. (Requirement 21-0404)

A. For ML-1 and ML-2 software, software designer and computer programmers (coders) must be provided with qualifications commensurate with the risk associated with the software.

B. The software design and the computational sequence necessary to meet the software requirements must be documented. This documentation should include (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.

C. Verification and validation (V&V) documentation, including test plans, interim test reports, and design review documentation, is typically produced concurrently with the design. See SOFT-V&V.

D. Deliverables consist of drawings, other descriptive documents, or both.

Typical design deliverables include the following:

- Software system architectural design[3]
- Software requirements specification (SWRS)[3]
- Software architectural design[3]
- Software detail design[3]
- Software system hazard analysis and mitigation documents (SWHA)[3]
- Software design traceability documentation (e.g., SWTM)[3]
- Computer program listings, including the program (design product)[3]
- O&M instructions
- SOFT-GEN deliverables[4]
- Other software design deliverables[5,6]

See ESM Chapter 8 for SSC software-related deliverables (e.g., logic diagrams, loop diagrams, point list tables) specific to instrument and control (I&C) system designs.

E. It is important to ensure that the design is defined, controlled, and verified; the design interfaces are identified and controlled; and design adequacy is verified by individuals other than those who designed the software. The eventual system engineer, users, or both should work with the design/development team to ensure documentation is completed to an appropriate level commensurate with the ML of the software.

[3] May be combined in one or more documents or provided separately. SWHA is not mandatory, but good practice; see SOFT-DESIGN Appendix A.
[4] Required and graded based on the nature, complexity, and associated ML per direction from the LANL SRLM.
[5] Includes software list, Form 2033, software data sheet, software baseline.
[6] As determined by LANL SRLM or as required in other ESM chapters (e.g., HMI graphical layouts/slides). See SOFT-V&V for associated V&V deliverables.

4.2 Software Requirements Specification

Refine software system requirements that are allocated to software into detailed software-level requirements. When complexity warrants, document these software-level requirements in a standalone SRS. Form 2323 is available for this purpose, and an SWRS document number for an SRS may be obtained from CoE (SharePoint). See Institute of Electrical and Electronics Engineers (IEEE) Std 830, *IEEE Recommended Practice for Software Specifications*, for guidance. An SRS is a valuable input to the V&V testing planning.

4.3 Software System Architectural Design

4.3.1 Applicability

This section deals primarily with the hardware and network architecture that the software integrates with and runs upon. As such, it is mainly applicable to SSC software, although non-SSC software that relies upon specialized hardware may be subject to these requirements and guidance. This section does not apply to SSC software that is designed to run upon existing hardware and that requires no hardware configuration changes.

4.3.2 Requirements

A. Develop the software system architectural design. Allocate system-level requirements to hardware, software, and interfaces of the system.

B. Sufficiently detail the system architectural design such that a person technically qualified in the subject can verify and validate that the design satisfies requirements without recourse to the designer.

C. For ML-1, ML-2, and some ML-3 systems for which software operational complexity and life cycle maintenance can be disadvantageous, evaluate and, as appropriate, employ designs that use no or minimal software.

D. Consider the following concepts, as applicable, in the system architectural design.

1. Isolation: Separate critical elements from each other to preclude undefined 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, and inputs/outputs to physically separate the safety elements from non-safety elements.

2. Independence/diversity: For ML-1 and ML-2 applications, employ independent, diverse systems in which the stimuli originate from and are handled by separate elements with different designs, independent hardware inputs, independent software modules, or a combination of these.

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 the software system such that it remains in a safe mode upon failure without compromising isolation features or other safety functions. Measures to mitigate the consequences of problems, as identified through analysis, shall be an integral part of the design. These potential problems include external and internal abnormal conditions and events that can affect the computer program. SOFT-DESIGN Appendix A provides guidance on analyzing

software to help ensure the necessary safety functions are met. (Requirement 21-0405)

4. Incompatibility/longevity: Design the system to ensure integrated hardware-software compatibility as much as possible throughout the life cycle of the system.

Table 21.3-4 Software Design Output Deliverables	
No.	Deliverable
01	Software system architectural design[2]
02	Software requirements specification[2]
03	Software architectural design[2]
04	Software detail design[2]
05	Software system hazard analysis and mitigation documents (SWHA; optional)[2]
06	Software design traceability documentation (e.g., SWTM)[2]
07	Computer program listings, including the program (design product)[2]
08	O&M instructions
09	SOFT-GEN deliverables[3]
10	Other software design deliverables[4, 5]

Note: Required for all software ML-1 through -4 or other ESM chapters.
[2] May be combined in one or more documents or provided separately.
[3] Required and graded based on the nature, complexity, and associated ML per direction from the LANL SRLM.
[4] Includes software list, Form 2033, software data sheet, software baseline.
[5] As determined by LANL SRLM or as required in other ESM chapters (e.g., HMI graphical layouts/slides). See SOFT-V&V for associated V&V deliverables.
[6] See ESM Chapter 8 for software-related deliverables (e.g., logic diagrams, loop diagrams, point list tables) specific to I&C system designs.

4.4 Software Architectural Design

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

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.

D. Where appropriate 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 would cause an unsafe system state.

E. Where appropriate 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 impact on the system's safe operation.

4.5 Software Detailed 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 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 and (b) allow computer programmers to code the design.

4.6 Software Design Traceability and Testability

Document traceability from the software design requirements to the software design deliverables. An SWTM may be used to document traceability. Other methods, however, such as tracing the requirement within the design deliverable, are also acceptable. Requirements tools and databases may be used.

4.7 Operations and Maintenance Instructions

Develop and provide O&M instructions for using and maintaining the computer program(s). See SOFT-MAINT for use and maintenance requirements and address these requirements in the O&M instructions, the software data sheet, or both.

5.0 SOFTWARE COMPUTER PROGRAM LANGUAGE (CODE)

A. Translate the software design into computer program language (code). (Requirement 21-0406)

B. For software, use standards or conventions as specified and described in the SOW or requirements.

C. Provide executable code as deliverables.

D. Provide a printout of the program source code listings.

E. Ensure code is developed with sufficient detail and clarity to allow someone technically qualified in the programming language to review it, understand it, and verify that it meets the software design requirements.

F. Ensure most code lines have explanatory notes/comments (pseudo code) to support future troubleshooting, bug fixes, and system modifications.

G. Employ configuration management and problem reporting/corrective management in accordance with SOFT-GEN.

6.0 APPENDIX

Appendix A, Software System Hazard Analysis and Mitigation (Guidance)

Appendix A. Software System Hazard Analysis and Mitigation (Guidance)

Appendix A provides guidance on analyzing software to help ensure the necessary safety functions are met.

A. For ML-1 through ML-3 software, consider conducting and documenting 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 SSC 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).*

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.)

If a standalone SWHA is desired, the document number may be assigned here.

CONTENTS

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 mandatory. Added acquisition document review and an example request package checklist; clarifications	Tobin Oruch, *ES-DO*	Lawrence Goen, *ES-DO*
2	02/11/2026	Merged/streamlined requirements for designed SSC and Non-SSC software. Clarified requirements for toolbox code V&V and Simple and Easily Understood software.	Tobin Oruch, *ES-FE*	Michael Richardson, *ES-DO*

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

SOFT-V&V: SOFTWARE VERIFICATION AND VALIDATION

1.0 INTRODUCTION

1.1 Purpose

SOFT-V&V addresses software verification and validation (V&V) and software approval for use (SWAU).

For non-SSC (structure, system, and component) software, it also includes such requirements of both safety basis toolbox codes used for analysis and “simple and easily understood” software used in the design of the SSCs.

1.2 Applicability

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

For SSC software, use SOFT-V&V in conjunction with ESM Chapter 15, *Commissioning*; AP-341-801, *Post Modification/Post Maintenance (PMT)*; or both.

> *Note: Throughout this chapter, the terms commission, commissioning, commissioning documents, Cx, etc., refer to either of the above test-related documents, based on applicability.*

When making software changes during V&V see SOFT-GEN for additional requirements.

1.3 Summary Tables

The following tables provide an overview of the requirements and deliverables of this Section. **They also address who performs tasks, and when.**

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[1]			Implementation Detail			Reference ESM Ch. 21 Ref.[5]
		1, 2	3	4	How	When[2]	Who[3, 4]	
1.0 For Designed SSC Software								
1	Review design (input) requirements	R	Gr	Gr	▪ Review design input documents ("docs") to SOFT-DESIGN design (input) requirement criteria ▪ As applicable to ML-1 and ML-2 software, ensure commercial grade dedication (CGD) requirements are addressed	▪ Before software design as much as practical	▪ SD (R) ▪ SRLM (R) ▪ LCxA (R) ▪ FDAR (R)	SOFT-V&V, 4; SOFT-DESIGN, 3
2	Develop/review test plan (SWTP)	R	Gr	Gr	▪ Develop software test plan (SWTP) and submit 60%, 90%, and 100% test plan submittals ▪ Review test plan according to SOFT-V&V test plan criteria ▪ Integrate into/with SSC commissioning (Cx) docs	▪ After requirements and concurrent with design ▪ Before interim testing, acceptance testing, or both	▪ SD (D, R, A) ▪ LCxA (R) ▪ SRLM (R, A)	SOFT-V&V, 4
3	Review software design	R	Gr	Gr	▪ Review according to SOFT-DESIGN and SOFT-V&V criteria ▪ Review 60%, 90%, and 100% software design documents/submittals	▪ Per software design schedule ▪ Before acceptance testing	▪ SD (R) ▪ SRLM (R) ▪ IR (R)	SOFT-V&V, 4; SOFT-DESIGN, 3, 4
4	Review computer program code	R	-	-	▪ Review computer program code where feasible	▪ Before acceptance testing	▪ SD (R) ▪ SRLM (R) ▪ IR (R)	SOFT-V&V, 4
5	Test/commission and review	R	Gr	Gr	▪ Test (including interim tests) per SSC design, software test plan, and Cx docs ▪ Perform acceptance test in simulated test environment (i.e., test bed) ▪ Perform acceptance test in operating environment ▪ Provide, review, and approve test (qualification) report (SWTR)	▪ Per software test plan, Cx docs, and SSC design ▪ Before approval for use	▪ T (D, R, A) ▪ SD (R, A) ▪ SO (R, A) ▪ LCxA (R, A) ▪ SRLM (R, A)	SOFT-V&V, 4

6	Review/SWAU	R	Gr	-	▪ Prepare software approval for use (SWAU) request package ▪ Review according to SWAU criteria ▪ Approve and document SWAU	▪ Before intended use	▪ SO (D) ▪ SRLM (R, A) ▪ FDAR	SOFT-V&V, 4; SOFT-MAINT, 3

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[1]			Implementation Detail			Reference ESM Ch. 21 Ref.[5]
		1, 2	3	4	How	When[2]	Who[3, 4]	
2.0 For Read-Only SSC Software								
1	Review (input) requirements	R	Gr	Gr	▪ Review SSC design/acquisition requirement (input) docs. (e.g., statement of work [SOW]) per governing doc control process ▪ As applicable to ML-1 and ML-2 software, ensure CGD requirements are addressed	▪ Before acquisition	▪ Per governing doc control process	SOFT-V&V, 6; SOFT-DESIGN, 3; SOFT-ACQUIRE, 2
2	Develop and review test plan	R	Gr	Gr	▪ Use SSC Cx documents as software test plan documents	▪ Per ESM Chapter 15 or AP-341-801, Post Modification/Post Maintenance ▪ Before Cx	▪ Per ESM Chapter 15 or AP-341-801, Post Modification/Post Maintenance	SOFT-V&V, 6
3	Review design	R	Gr	Gr	▪ Review software as part of the SSC design deliverables (ESM Chapter 1)	▪ Per SSC design schedule	▪ Per governing SSC review process	SOFT-V&V, 6
4	Commission and review	R	Gr	Gr	▪ Test per SSC design and Cx docs ▪ Provide, review, and approve Cx report	▪ Per Cx docs and SSC design ▪ Before approval for use	▪ Per ESM Chapter 15 or AP-341-801, Post Modification/Post Maintenance	SOFT-V&V, 6
5	Review and approve for use	R	Gr	-	▪ Prepare SWAU request package ▪ Review according to SWAU criteria ▪ Approve and document SWAU	▪ Before intended use	▪ SO (D) ▪ SRLM (R, A)	SOFT-V&V, 6; SOFT-MAINT, 3

[1] ML = management level; R = required; Gr = required but graded; "-" = not required

[2] "When" constitutes the V&V control points (points in the process when reviews are performed)

[3] D = develop or execute; R = review; A = approve

[4] 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

[5] Ref. = ESM Chapter 21 section name and subsection number (e.g., Section SOFT-GEN, Subsection 2, *Software Identification and Determination*)

Table 21.4-2 SOFT-V&V 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[1]			Implementation Detail			Reference ESM Ch. 21 Ref.[5]
		1, 2	3	4	How	When[2]	Who[3, 4]	
1.0 For Designed Non-SSC Software								
1	Review design (input) requirements	R	R	R	▪ Review design input docs according to SOFT-DESIGN design (input) requirement criteria ▪ As applicable to ML-1 and ML-2 software, ensure commercial grade dedication (CGD) requirements are addressed	▪ Before software design as much as practical ▪ Before software acquisition	▪ SD (R) ▪ SRLM (R)	SOFT-V&V, 4; SOFT-DESIGN, 3
2	Model V&V	R	Gr	Gr	▪ As applicable, V&V the appropriateness of the model in addition to the computer code ▪ Document the model V&V in design and testing reviews	▪ After requirements and concurrent with design	SRLM (R, A)	SOFT-V&V, 4; SOFT-DESIGN, 4
3	Develop/ review test plan	R	Gr	-	▪ Develop SWTP (e.g., Form 2329 and Form 2328, or Form 3055 if preferred/available) and submit 60%, 90%, and 100% test plan submittals ▪ Review test plan according to SOFT-V&V test plan criteria	▪ After requirements and concurrent with design ▪ Before formal interim testing, acceptance testing, or both	▪ SD (D, R, A) ▪ SRLM (R, A)	SOFT-V&V, 4
4	Review software design	R	Gr	Gr	▪ Review according to SOFT-DESIGN design and SOFT-	▪ Per software design schedule	▪ SD (R) ▪ SRLM (R)	SOFT-V&V, 4; SOFT-DESIGN, 4

					V&V criteria ▪ Review 60%, 90%, and 100% software design documents/submittals	▪ Before acceptance testing	IR (R)	
5	Review computer program code	R	-	-	▪ Review computer program code where feasible	▪ Before acceptance testing	▪ SD (R) ▪ SRLM (R) ▪ IR (R)	SOFT-V&V, 4, 8 SOFT-DESIGN, 5
6	Test and review	R	Gr	Gr	▪ Test per software test plan ▪ Provide, review, and approve test report	▪ Per software test plan ▪ Before approval for use	▪ T (D, R, A) ▪ SD (R, A) ▪ SO (R, A) ▪ SRLM (R, A)	SOFT-V&V, 4
7	Review/ SWAU	R	Gr	-	▪ Prepare SWAU request package ▪ Review according to SOFT-MAINT SWAU criteria ▪ Approve and document SWAU	▪ Before intended use	▪ SO (D) ▪ SRLM (R, A)	SOFT-V&V, 4; SOFT-MAINT, 4
2.0 For Acquired Software								
1	Model V&V	R	Gr	Gr	▪ As applicable, V&V the appropriateness of the model in addition to the computer code ▪ Document the model V&V in acquisition and testing reviews	▪ Should be completed before acquisition; must be completed before SWAU	▪ SRLM (R, A)	SOFT-V&V, 5; SOFT-DESIGN, 7; SOFT-ACQUIRE (ALL)
2	Review acquisition (input) requirements	R	R	R	▪ Review acquisition input docs (e.g., SOW) according to SOFT-ACQUIRE acquisition (input) requirement criteria ▪ As applicable to ML-1 and ML-2 software, ensure CGD requirements are addressed	▪ Before software acquisition	▪ SO (R) ▪ SRLM (R)	SOFT-V&V, 5; SOFT-ACQUIRE (ALL)
3	Develop/ review test plan	R	Gr	-	▪ Develop SWTP (e.g., Form 2329 and Form 2328, or Form 3055 if preferred/available) ▪ Review test plan according to SOFT-V&V test plan criteria	▪ After requirements and concurrent with acquisition ▪ Before acceptance testing	▪ SD (D, R, A) ▪ SRLM (R, A)	SOFT-V&V, 5
4	Review acquisition	R	Gr	-	▪ Review and document conformance with acquisition documents	▪ Before acceptance testing	▪ SO (D)	SOFT-V&V, 5
5	Test and review	R	Gr	Gr	▪ Test per software test plan ▪ Provide, review, and approve test report	▪ Per software test plan ▪ Before approval for use	▪ T (D, R, A) ▪ SD (R, A) ▪ SO (R, A) ▪ SRLM (R, A)	SOFT-V&V, 5

6	Review/ SWAU	R	Gr	-	▪ Prepare SWAU request package ▪ Review according to SOFT-MAINT SWAU criteria ▪ Approve and document SWAU	▪ Before intended use	▪ SO (D) ▪ SRLM (R, A)	SOFT-V&V, 5; SOFT-MAINT, 4
3.0 For Acquired Non-SSC (OTS)—Toolbox Codes								
1	V&V and SWAU	R	R	R	▪ V&V per the registry instructions for the toolbox code ▪ Prepare SWAU request package ▪ Review according to SOFT-MAINT SWAU criteria ▪ Approve and document SWAU	▪ For each toolbox code installation ▪ Before intended use	▪ SO (D, R, A) ▪ SRLM (R, A)	SOFT-V&V, 7; SOFT-MAINT, 4
4.0 For Simple and Easily Understood Non-SSC Software Used in Design of SSCs								
1	V&V as part of SSC design V&V	R	Gr	Gr	▪ Pre-verify using the methods previously described for non-SSC software, or V&V the software each time it is used for SSC design in accordance with the SSC design process ▪ Provide and review minimum software information (e.g., computer-generated evidence of the programmed algorithms or equations) ▪ Apply appropriate SSC design review rigor to ensure the computer program produces satisfactory results	▪ Each time the software is used for an SSC design and when required by the SSC design V&V process ▪ Before intended use	▪ As per the applicable SSC design V&V process	SOFT-V&V, 8; SOFT-MAINT, 4

[1] ML = management level; R = required; Gr = required but graded

[2] "When" constitutes the V&V control points (points in the process when V&V activities are performed)

[3] D = develop or execute; R = review or inspect; A = approve

[4] SD = software designer; SO = software owner; SRLM = software responsible line manager; FDAR = facility design authority representative; T = testers; IR = independent reviewer

[5] An SD subject matter expert (SME) other than the SD who developed the code shall perform the code review

[6] Ref. = ESM Chapter 21 section name and subsection number (e.g., Section SOFT-GEN, Subsection 2, *Software Identification, Ownership, and Determination*)

2.0 DEFINITIONS AND ABBREVIATIONS

Selected definitions specific to SOFT-V&V follow.

Other Chapter 21 terms are in Appendix A of SOFT-INTRO. Beyond that, many definitions are contained in GLOS-COE-1, the *Conduct of Engineering Glossary*, and those definitions supersede should they conflict with Chapter 21.

- *Verification* is the act of reviewing, inspecting, testing, checking, auditing, or otherwise determining and documenting whether items, processes, services, or documents conform to specified requirements (Ref. NQA-1). In the context of software, verification means providing objective evidence that the software and its associated products conform to requirements (e.g., for correctness, completeness, consistency, accuracy) for all life cycle activities during each life cycle process (e.g., acquisition, supply, development, operation, 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).
- *Software design verification* is the process of determining if the product of the software design activity fulfills the software design requirements (NQA-1-2008/1a-2009).
- *Validation* is 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. NQA-1). In the context of software, validation means providing evidence that the software and its associated products satisfy system requirements allocated to software at the end of each life cycle activity, solve the right problem (e.g., correctly model physical laws, implement business rules, use the proper system assumptions), and satisfy the intended use and user needs.
- *Software approval for use (SWAU)* is an approval that confirms that the software requirements have been satisfied (including installation and operating instructions) and that the software is ready to be used in the intended operating environment.
- *Simple and Easily Understood (Non-SSC) software* is that which that satisfies the following criteria:
  - The software is used in the design of SSCs;
  - The results of the computer program can be easily confirmed through hand calculations;
  - A person technically qualified in the subject can review and understand the program and the supporting calculations; and,
  - The software is individually verified with each use (e.g., calculation). (Based on NQA-1).

  Note: When such is present, and the "Each-Time" V&V method is chosen, users should follow that subsection herein in lieu of other V&V requirements herein.

3.0 GENERAL REQUIREMENTS

The following requirements apply to all software (Requirements 21-0501 – 0508):

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

Note: For paragraphs B–E, the documentation details are specified in subsequent subsections.

B. Trace, document, and V&V the software from requirements development through design (as applicable), acquisition (as applicable), testing, and SWAU.

C. Document the V&V results, including the identification of the verifiers and validators.

D. For the review of V&V activities for ML-1 and ML-2 software, 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 the review of V&V activities for ML-3 and ML-4 software, document the reviewers and their comments and retain reviewer comments as records per the record retention schedule.

F. Ensure the limits of use associated with the V&V are clearly defined.

G. For other than acceptance testing reviews, if the review alone is inadequate to determine whether requirements are satisfied, use alternate calculations or develop and integrate (interim) tests to support the review.

H. Suppliers that design software specifically for LANL must V&V software based on subsection 4.0 on designed software. The V&V must be performed 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 and retention (ref. SOFT-INTRO, §2.3).

I. Ensure the software identification number (SWID) is part of the V&V documentation record number.

J. Perform software design verification in accordance with the requirements of this document and of AP-341-620, *Review and Verification of Design Documents*. For SSC software, document software design verification [e.g., using Form 2329, Software Test Plan and Report (SWTPR) attached to (when new) form SOFT-V&V-FM01 or, for approved/operating software modifications, SOFT-GEN-FM04].

K. Software design verification shall be performed by competent individual(s) or group(s) other than those who developed and documented the original design.

1. Verification may be performed by the originator's supervisor, if (a) the supervisor did not specify a design approach, rule out design considerations, or establish the design inputs or (b) the supervisor is the only individual in the organization competent to perform the verification.

L. Verification must include one or a combination of design reviews, alternate calculations, and tests performed during computer program development.

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.

A form that may be used for documenting test cases: 2328 - Software Test Case (SWTC)

In addition, examples are on the website under SOFT-V&V (useful content while perhaps not using above forms).

4.0 DESIGNED SOFTWARE V&V AND SWAU

This subsection is for software designed by or for LANL. When "simple and easily understood" software such as a spreadsheet, read Subsection 8.0 first.

Table 21.4-3 Minimum Required V&V Activities—Designed Software

4.0 § No.	V&V Activity Description
4.1	Design requirement (input) review
4.2	Software test plan development and review

Table 21.4-3 Minimum Required V&V Activities—Designed Software	
4.0 § No.	V&V Activity Description
4.3	Software design review
4.4	Software code review
4.5	Testing/commissioning and review
4.6	Software review and approval for use

4.1 Design Requirement (Input) Review

Table 21.4-3 above lists the minimum V&V and SWAU activities that must be performed for developed software. Additional details on these activities is provided later in this subsection.

Review design requirement inputs to ensure they satisfy the design requirement criteria in SOFT-DESIGN and maintain a record of the review per the governing document control process. The SWAU (or, for changes, SWCP or SWNCP) is used to document the completion of this review.

- Perform the review before software design where possible.
- For ML-1 through ML-3 SSC software, the SD, SRLM, LCxA rep (optional), and FDAR must perform the review. FDAR review is not required for ML-4 software.
- For non-SSC software, the SD and SRLM must perform the review.

Note: SOFT-DESIGN steps must be followed in the design phase and considered throughout the V&V phase. See also SOFT-DESIGN Appendix A when designing test cases.

4.2 Software Test Plan Development and Review

Plan and execute tests to verify conformance of the computer program(s) to requirements. The SWTP may include interim testing to support software design activities, acceptance testing to support commissioning (SSC only), in-use testing to support operations after SWAU, or a combination of these. For SSC software, the acceptance testing portion of the SWTP must be integrated into or with commissioning documentation.

NOTE: SOFT-V&V Appendix A, Computer Program Test Plan Requirements (Table), contains detailed test plan expectations in addition to those in this subsection.

Form 2329 - Software Test Plan and Report (SWTPR) may be used to produce the plan; also, one may produce a combined Plan and test cases using 3055 - Software Test Plan (SWTP) Template if preferred and available (internal links).

Develop the SWTP after software requirements have been established and concurrently with the software design. Submit the final test plan with the final software design and before acceptance testing.

The SD develops (and reviews and approves along with others) the SWTP. (Where possible, the SD should be a subject matter expert other than the SD who designed the software.)

- For ML-1 and ML-2 software**,** SWTP review by an SQA SME is recommended.

Address the following in the test plan:

A. For SSC software, plan tests as required by the SSC design; ESM Chapter 15, *Commissioning* (as applicable)*;* and this chapter. Document the plan in an SWTP. Review and disposition comments on the SWTP in accordance with the governing document control process.

   For ML-1 software, perform at least one acceptance test in a simulated environment (i.e., test bed or bench testing). For ML-2 software, perform at least one acceptance test in a

simulated environment when required by the FDAR. When performing acceptance tests in the simulated environment, do so before performing acceptance tests in the actual operating environment.

When the simulated environment without hardware cannot adequately test the software, as determined by the FDAR, test in a simulated environment with hardware, or without associated hardware if possible.

B. Perform a comprehensive acceptance test in the operating environment before use.[1] Include support software tests as appropriate. Refer to SOFT-INTRO for tester qualifications (e.g., for ML -1 or 2).

As required by other ESM chapters, plan for interim testing during planned stop points in the software design process (in addition to the final acceptance test). Interim tests may be formal or informal to support reviews.

Test plan documentation is required for formal interim tests used to support reviews, and this documentation must satisfy the same criteria as the acceptance test. Include the tests and test results in the acceptance testing documentation.

C. For ML-1 through ML-3 software, include in-use tests to be performed after SWAU to verify continued acceptable performance in the operating environment. Perform in-use tests whenever the computer program is installed on a different computer or when there are significant changes in the operating system.

D. Create an SWTP record form or template that records the following (as applicable).

1. SSC(s), software tested, or both
2. Date of test
3. Characteristics tested
4. Tester, data recorder, or both
5. Type of observation (i.e., type of test)
6. Test results, applicability, and acceptability
7. Action taken in connection with any deviations
8. Reference to the test methodology/test plan used with associated revision
9. Test result evaluator/acceptor
10. A unique test record identifier to allow traceability to the specific item tested (if not subsequently assigned as part of the records management protocols)
11. Test environment (e.g., computer hardware, operating system)
12. Measuring and test equipment used (include equipment identification, calibration date, and calibration expiration date)
13. Simulation models used
14. Test problems/cases [2328 - Software Test Case (SWTC) may be used]

E. Provide 60%, 90%, and 100% test plan submittals for review and approval. Review, approve, and control the test plan in accordance with the governing document control process.

[1] Also meet NQA-1 Req 11, 200 Test Reqs, (c): “If temporary changes to the approved configuration of a facility are required for testing purposes, approval by the design authority is required prior to performing the test. Note, when testing in the operating environment introduces unacceptable risks, testing in an identical beta environment is acceptable, with FDAR approval.

F. Identify any required SWTP hold or witness points, and associated responsibilities, or both.

G. For ML-1 and ML-2 software, design tests to demonstrate that performance requirements are satisfied and unintended results are detected (to the extent possible). Consider negative testing in addition to positive (see Definitions).

H. Identify authorized software testers. Note: Testers and reviewers should normally be identified by role rather than by name.

I. Identify individuals responsible for reviewing and approving the completed software test. As a minimum, the LANL SO and SRLM review and approve the SWTP.

J. Produce Software Test Report (SWTR).

- For ML-1 through ML-3 software, SWTR reviewers/approvers must include the lead tester (SD or SME familiar with the design detail and intended use of the computer program), LANL SO, LCxA (SSC only, optional), and LANL SRLM.
- For ML-4 software, SWTR reviewers/approvers must include the LANL SO, LCxA (SSC only, optional), and LANL SRLM.

4.3 Software Design Review

Review the software design to ensure that it satisfies the software design criteria specified in SOFT-DESIGN and maintain a record of the review per the governing document control process. Perform the following as part of the design review (Requirements 21-0509 – 0519):

- Verify the acceptability of the design requirements.
- Evaluate the technical adequacy of the software design approach.
- Ensure internal completeness, consistency, clarity, and correctness of the software design.
- Verify that the software design is compliant with and traceable to the approved software design requirements.
- Review any interim test results.

Review the design in accordance with the software design deliverable submittal schedule and before acceptance testing.

The SD and SRLM review the software design. In addition, for ML-1 and ML-2 software, one or more individuals other than those who designed the software must review the design. Independent reviewers may be from the same organization.

4.4 Computer Program Code Review

Review the computer program code to ensure that it satisfies the criteria specified in SOFT-DESIGN and maintain a record of the review per the governing document control process. Review the code before acceptance testing.

The SD and SRLM review the code. In addition, an independent SME other than the SD who developed the code performs a review.

4.5 Acceptance Testing/Commissioning, Software Test Report, Review

Perform acceptance testing in accordance with approved design documents, SWTPs, commissioning documents, or a combination of these. Following acceptance testing, prepare SWTRs and perform SWTR reviews before SWAU. Ensure the following items are addressed (Requirements 21-0520 – 0522):

A. Complete activities to prepare for testing in accordance with any governing work control documents and address the following items.

1. Ensure required software backups are accessible should a need to “roll back” to original files arises.
2. Ensure that documentation required before testing has been received.
3. Review the results of prior testing and resolution of any identified deficiencies (e.g., bug or error reports). Ensure that issues have been sufficiently resolved to proceed with the test.
4. Ensure that those performing or supporting testing have all required training, qualifications, and authorization and include all individuals specified in the SWTP.
5. Ensure that those performing or supporting testing have a clear understanding of their roles and responsibilities and the SSCs that may be impacted by the tests.
6. For SSC software, ensure specified calibration, environmental conditions, and storage requirements are maintained throughout the test process.
7. For SSC software, ensure any necessary measuring and test equipment (e.g., thermometers, manometers) is calibrated and on hand. (Requirement 21-0523)
8. Control Testing: For SSC software, indicate status (e.g., tested, not tested, testing in process) on the SSC or in documents traceable to it to prevent inadvertently operational use (e.g., segregate the SSC or use tags, markings, shop travelers, stamps, and/or inspection records). List the authority for removal/change of status. For Non-SSC software, keeping in a test environment until SWAU is sufficient. (Requirement 21-0524)
9. Conduct tests under controlled conditions using approved acceptance and performance criteria.
10. Retest (e.g., regression test) modified or replaced software.
11. For SSC software, ensure the required post-test configuration is in place after completing the test.

B. Document the test results on the approved SWTP record forms, templates, or both.

C. Review and evaluate the test results and verify that all requirements have been satisfied and that the software produces correct results. Perform the following as part of the test results review:

1. Perform preliminary evaluations to determine the validity of the test results and the need for further testing. Identify any additional test requirements and required changes to the test plan and retest as needed.
2. Analyze and evaluate the data to verify the completeness of results, conformance to design requirements, achievement of test objectives, and conformance to the SWTP.

D. Document the acceptability of the test results in the SWTR.

E. Ensure that the SWTR is reviewed by all individuals specified in the SWTP.

F. Retain test documentation per the governing document control/records management processes. Ensure test records have a unique identifier and are traceable to the tested software.

4.6 Review and Approval for Use

Before using software for its intended purpose, obtain a formal software approval for use. For initial SWAU, prepare a SWAU request package and document approval via Chapter 21 form SOFT-V&V-FM01, *Software Approval for Use* (Requirement 21-0525).

A. The SWAU request package shall include all documentation necessary to demonstrate that the software is ready for its intended use, including

- satisfactory completion of V&V;
- attestation that any applicable operational and maintenance controls from SOFT-MAINT are in place; and
- for ML-1 through ML-3 software, documentation of any associated limitations, access controls, etc., for software use.

B. SWAU following changes to existing and approved software shall be documented as follows:

- For ML-1 through ML-3 software, use Chapter 21 forms SOFT-GEN-FM03, *Non-SSC Software Change Package Form (SWNCP),* or SOFT-GEN-FM04, *SSC Software Change Package Form (SWCP)*.
- For ML-4 SSC software, SWAU is implicitly granted following successful completion of commissioning or post-modification testing per AP-341-801, *Post Modification/Post Maintenance Testing*.
- For ML-4, non-SSC software, when used by Facility Engineering divisions, SWAU is implicit when software is in the Graded Software Inventory/GSI as “Completed” and current on GSI review date, indicating SRLM concurs with its use.

5.0 ACQUIRED SOFTWARE V&V AND SWAU

Table 21.4-4 lists the minimum V&V and SWAU activities that must be performed for acquired software, both SSC and Non-SSC. Additional detail on these activities is provided later in this subsection.

Note: Read-only SSC (firmware) software requirements are contained in Subsection 6.0.

Table 21.4-4 Minimum Required V&V Activities—Acquired Software

5.0 § No.	V&V Activity Description
5.1	Acquisition requirement (input) review
5.2	Test plan development and review (optional for ML-4)
5.3	Acquisition review (optional for ML-4)
5.4	Testing and review
5.5	Review and approval for use

5.1 Acquisition Requirement (Input) Review

Before software acquisition and in accordance with the governing acquisition/procurement and associated document control processes, the SO and SRLM shall review the acquisition/procurement documents (e.g., statement of work, Exhibit H) to ensure that they satisfy the acquisition criteria in SOFT-ACQUIRE and that the associated software requirements are correct and complete.

As applicable, ensure CGD requirements are addressed (Requirement 21-0526).

5.2 Test Plan Development and Review

For ML-1 through ML-3 software, follow the requirements of Subsection 4.2 with the following exceptions:

A. For software obtained from an IESL-listed supplier, prepare a simplified test plan that uses the applicable supplier-provided test cases to demonstrate that the software is adequately installed, suitable for the intended application, and properly working in the intended operating environment.

- Acceptance testing in a simulated environment (i.e., test bed) does not apply.
- Testing associated with software design (e.g., interim tests) does not apply.

B. Preparation or incorporation of a hazard analysis is not required. Supplier-provided test cases, documentation, and comparative alternate calculations/published data may be used to demonstrate that the software meets requirements for its intended use.

C. The SO or an SD may develop the test plan.

5.3 Acquisition Review

For ML-1 through ML-3 software, before acceptance testing, the SO shall 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 the software in accordance with CGD requirements. As applicable, verify valid software licenses and markings that indicate an original manufacturer's product to prevent acquisition of suspect/counterfeit software. Document and retain a record of the review in accordance with the governing document control/records management processes.

5.4 Testing and Review

A. For ML-1 through ML-3 software, follow the requirements of Subsection 4.3 except for those associated with software design (e.g., interim testing).

B. For ML-4 software, install and test the software per the supplier instructions.

5.5 Review and Approval for Use

Follow the requirements of Subsection 4.6.

6.0 READ-ONLY SSC SOFTWARE V&V AND SWAU

Read-only SSC software is firmware/embedded software that cannot be changed except through replacement of the computer program, the associated SSC, or both (see SOFT-INTRO for further discussion of it).

- If such software meets all conditions for partial exclusion from Chapter 21 in SOFT-INTRO (§2.3), and the required actions are performed, neither this subsection nor the balance of the chapter are required.
- If, however, any partial exclusion entry condition for the applicable ML is not met, perform Subsections 6.1–3.

6.1 Design Review

When LANL can be involved in the design of the software, review the design as part of the review of the SSC (e.g., at the phases of SSC design deliverable schedule in ESM Chapter 1 Z10 Att. C) under the governing SSC design review process.

6.2 Testing/Commissioning and Review

Test the read-only SSC software per the SSC design and commissioning documents. Provide, review, and approve a commissioning report before approval for use.

6.3 Review and Approval for Use

Follow Subsection 4.6 as applicable to read-only SSC software.

7.0 TOOLBOX CODE V&V AND SWAU

Note: These are programs used by Safety Basis analysts.

A. Verify and validate the toolbox codes (see SOFT-INTRO Appendix A for definition and information) in accordance with the registry instructions for the specific toolbox code. Review and complete SWAU per Subsection 4.6.

B. Complete V&V and SWAU for each installation of a toolbox code. The SO performs these activities, and the SO and SRLM review and approve them.

8.0 "SIMPLE & EASILY UNDERSTOOD" NON-SSC SOFTWARE V&V/SWAU

For "simple and easily understood" non-SSC software used in the design of the SSCs, V&V/SWAU the software (all MLs):

- each time the software is used for an SSC design,
- when required by the SSC design V&V process, and
- before approval of the associated SSC design.

There are two options for doing this verification – either pre-verifying or verifying each time the software is used — as described below.

8.1 Pre-Verification[2]

A. Perform V&V/SWAU per Subsections 4 or 5.

8.2 Each-Time Verification Process[3]

A. To individually verify, do the following:

1. Follow the SSC design process. Document the design analysis (e.g., AP-341-605, *Calculations*, for LANL). Include the following in the calculation:

a. The objective of the analysis

[2] This might be most appropriate for a spreadsheet tool such as one to inform the need for lighting protection, wherein it is V&V'd, formulas then locked and published for users to enter specifics and get results.

[3] Here, the software is V&V'd by the calculation verifier each time, so likely more appropriate for software that won't be reused often, and/or has input values embedded in it by the developer/user (e.g., an unlocked spreadsheet).

b. Design inputs and their sources

c. Results of literature searches or other applicable background data

d. Assumptions and indication of assumptions that must be verified as the design proceeds

e. Identification of any computer calculation, including identification of the computer type; computer program name; 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

2. Review and approval. Include the following software information in the SSC design verification documentation and retain it as a record (e.g., calculation documentation):

a. Design inputs

b. The computer program–generated results

c. Computer-generated evidence of programmed algorithms or equations (e.g., computer program listings, spreadsheet cell contents)

3. Ensure that those performing individual software verifications are cognizant that software may not have undergone CGD and 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. They must show that the software produces correct solutions for the encoded mathematical model within defined limits for each parameter employed. As applicable, they need to show that the encoded mathematical model produces a valid solution to the physical problem associated with the application.

B. Regarding SWAU, following usage controls in SOFT-MAINT (e.g., for ES, PFE, PIE) ensure software adequacy.

9.0 RESOURCES

The resource collection posted with SOFT-V&V contains example V&V documentation and templates. Examples provided are for informational purposes only and do not always use the latest form available. These examples may include the following:

- Form 3056, *Software Requirements Traceability Matrix (SWRTM or SWTM)*
- Form 3055, *Software Test Plan (SWTP)*
- Form 2328, *Software Test Case (SWTC)*
- Form 2329, *Software Test Plan and Report (SWTPR)*
    - Example Requirements Traceability Matrix (Source Tracker)
- Example No. 1 Test Plan (Source Tracker)
- Example No. 1 Test Plan/Report (Source Tracker)
- Example No. 2 Test Report
- Example No. 3 Test Report
- Example SWAU Request Package Checklist (MACCS2)

10.0 APPENDICES AND FORMS

- SOFT-V&V Appendix A, Computer Program Test Plan Requirements (Table)
- SOFT-V&V-FM01: Software Approval for Use Form (SWAU); FM01(I) Instructions with Example

Posted with SOFT-GEN:

- SOFT-GEN-FM03: Non-SSC Software Change Package Form (SWNCP)
- SOFT-GEN-FM04: SSC Software Change Package Form (SWCP)

Institutional forms that may be used/adapted

- 2328 - Software Test Case (SWTC)
- 2329 - Software Test Plan and Report (SWTPR)

In addition, examples (not using above forms) are on the Chapter 21 webpage under SOFT-V&V.

Appendix A. Computer Program Test Plan Requirements

Appendix A provides computer program test plan requirements in addition to those above (e.g., in Subsection 4.2, which also addresses plan development role). The text in italicized print should be used as guidance for all items marked as "Gr." (Requirements 21-0527 – 0545)

No.	Requirement	Apply by ML[1]			
		1	2	3	4
	1.0 General Test Plan Requirements[2]				
1.01	Include or reference the test name, objective(s), and date.	R	R	R	*
1.02	Include or reference the test configuration *(including pre-test, during testing, and post-test configurations where appropriate).*	R	R	Gr[3]	*
1.03	Satisfy test prerequisites. Prerequisites include the following as applicable: calibrated instrumentation, appropriate and special equipment, trained personnel, the condition of the test equipment and the item to be tested, suitable environmental conditions, and provisions for data acquisition. Ensure the suitability and readiness of equipment and algorithms used for simulated computer program test inputs/fault seeding.	R	R	R	*
1.04	Ensure that suitable environmental conditions are satisfied during testing.	R	R	R	*
1.05	Ensure that adequate instrumentation is available and used.	R	R	R	*
1.06	Ensure that appropriate tests and equipment are used (e.g., 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 assess 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	Ensure that 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 *from or approved by the responsible design organization*.	R	R	Gr	*
1.11	Control the test. (This requirement includes controlling changes during testing and, as appropriate, comprises 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 that fulfills 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 an approved configuration management SSC of an operating facility are required for testing purposes, obtain approval by the facility design authority or FDAR and others as required in the governing work control process (e.g., AP-341-504, *Temporary Modification Control*) before 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	*
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	*

No.	Requirement	Apply by ML[1]			
		1	2	3	4
1.19	When special process examinations/tests are performed (e.g., nondestructive examination, 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 STD-342-100, *Engineering Standards Manual,* Chapter 13, *Welding, Joining & NDE*, and P330-5, *Special Processes.*	R	R	R	*
1.20	Perform testing in accordance with the governing work control process and the test plan and 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 the 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	Within applicable work control documents (e.g., field change, design change) that govern configuration management for the item, describe the process for obtaining consent to waive specified hold points (testing, inspection, witness holds, or a combination of these) or to make changes to the test plan. If the waiver and change processes are 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 before work continues beyond the designated hold point. Document the consent.	R	R	R	*
1.25	As applicable, develop and prepare an approach to sampling. *When sampling is used, base it on standard statistical methods with approval from the responsible engineer for the item. Obtain support from the responsible engineer, quality assurance representative, or both in development of the plan as required. When sampling is used for accepting commercial-grade items and services, base it upon standard statistical methods with supporting engineering justification. Consider lot/batch/traceability, homogeneity, and the complexity of the item.* 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	Gr	*
1.26	*Demonstrate the 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.[4]*	R	R	Gr	*
1.27	*Where the test is intended to verify only specific design features, verify the other features of the design by other means. Establish and verify scaling laws when tests are performed on models or mockups.[4]*	R	Gr	Gr	*
1.28	*Perform error analysis, where applicable, on the results of model test work before using the final design.[4]*	R	Gr	Gr	*
	2.0 Additional Test Plan Requirements for General Computer Program Testing[2]				
2.01	*Involve engineering, in accordance with the governing work control process and as applicable, in safety software inspection, including obtaining concurrence with the acceptability of safety software verification results.*	R	R	Gr	*
2.02	*Obtain test requirements and acceptance criteria from the organization responsible for the computer program's use.*	R	R	Gr	*
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	*Include criteria for establishing test cases.*	R	R	Gr	*
2.07	*Include requirements for testing logic branches.*	R	R	Gr	*
2.08	Include requirements for hardware integration.	R	R	R	*
2.09	Include anticipated output values.	R	R	R	*

No.	Requirement	Apply by ML[1]			
		1	2	3	4
2.10	Include acceptance criteria, reports, records, standard formatting, and conventions.	R	R	R	*
2.11	Demonstrate the computer program's adherence to documented requirements.	R	R	R	*
2.12	*Evaluate technical adequacy through the comparison of test results against results from alternative methods, such as hand calculations, calculations using comparable proven programs, or empirical data and information from technical literature.*[5]	R	R	Gr	*
2.13	*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.*[4]	R	R	Gr	*
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	*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.*[6]	R	R	Gr	*
2.16	For system software testing *(new or revised software), evaluate, review, test, and accept the software 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.*	R	R	Gr	*
	3.0 Additional Test Plan Requirements for Computer Program Acceptance Testing[7]				
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 and operating requirements).[8]	R	R	R	*
3.03	Demonstrate that the computer program properly handles abnormal conditions and events, *as well as credible failures. As applicable, include cybersecurity vulnerability testing.*	R	R	Gr	*
3.04	Demonstrate that the computer program does not perform adverse unintended functions.	R	R	R	*
3.05	*Demonstrate that the computer program does not degrade the system either by itself or in combination with other functions or configuration items.*	R	R	Gr	*
3.06	Ensure that acceptance testing is performed and passed before the computer program's approval for use.	R	R	R	*
3.07	Ensure that configuration items are identified and are under configuration change control before starting acceptance testing.	R	R	R	*
3.08	Ensure that acceptance testing is planned, performed, and passed *for all software design requirements.*	R	R	Gr	*
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 before use.	R	R	R	*
3.11	*Provide for the documentation and disposition of observations regarding unexpected or unintended results before test approval.*	R	R	Gr	*
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 verify that a modified system(s) or system component(s) still meets specified software design requirements.	R	R	R	*

No.	Requirement	Apply by ML[1]			
		1	2	3	4

Notes:
[1] R = required; Gr = graded by using *italicized text* as guidance and non-italicized text as required; * = as required by ESM Chapter 15, *Commissioning,* the SRLM, or both
[2] 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.
[3] Not used.
[4] Applies to design verification testing only; reference NQA-1 Part I Requirement 3, *Design Control*.
[5] Applies to non-SSC software only.
[6] Apply as required during testing to ensure that test results are valid.
[7] 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.
[8] If the software is required to perform a technical safety requirement surveillance, demonstrate satisfactory execution of the surveillance as part of the acceptance test.

CONTENTS

RECORD OF REVISIONS

Rev.	Date	Description	POC	RM
0	06/23/2016	Initial issue	Tobin Oruch, *ES-DO*	Mel Burnett, *CENG-OFF*
1	05/25/2017	Clarifications for existing (in-use) software, other minor clarifications	Tobin Oruch, *ES-DO*	Lawrence Goen, ES-DO
2	02/11/2026	Minor streamlining and updating.	Tobin Oruch, *ES-FE*	Michael Richardson, *ES-DO*

SOFT-MAINT: Use, Maintain, & Retire

1.0 INTRODUCTION

1.1 Purpose and Applicability

This section describes the requirements for using, maintaining, and retiring software.

2.0 DEFINITIONS AND ACRONYMS

Appendix A of SOFT-INTRO addresses defined Chapter 21 terms.

3.0 SSC SOFTWARE

3.1 Use and Maintain SSC Software

Control management level (ML)-1 through ML-4 software to ensure that the use remains consistent with the software design and approved acceptance test results. For ML-1 through ML-3 software, in addition to the applicable controls from other chapter sections, ensure the following controls are established and documented on the software datasheet (SDWS) and otherwise as allowed by this chapter before software approval for use (SWAU). For existing software (also referred to as in-use or legacy software) that is noncompliant, initiate an IM action to resolve deficiencies (see SOFT-GEN for details). (Requirement 21-0601)

Use and maintain the software using these controls until the software is retired.

- Application documentation
- Access control specifications
- Computer system vulnerability protections
- Problem reporting and corrective actions
- In-use tests
- Configuration management
- Inventory
- Supporting quality assurance program controls

Process and retain documentation in accordance with the governing document control and records management process indicated by the software responsible line manager (SRLM), described in the SWDS, or both. Ensure the correct software identification number (often obtained before completing Form 2033) is part of the documentation. See AP-341-402, *Engineering Document Management in Operating Facilities,* for details.

The following summary table provides an overview of the requirements and deliverables of this Section. **It also contains details on who performs related tasks, and when**

Table 21.6 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[1]			Implementation Detail			ESM Ch. 21 Ref.[5]
		1, 2	3	4	How	When	Who[2]	
1	Use and maintain the software	R	Gr	Gr[3]	▪ Use and control the software to ensure that the use remains consistent with the software design and approved acceptance test results ▪ Maintain software application documentation ▪ Maintain licenses and registrations ▪ Use and maintain access controls ▪ Use and maintain vulnerability protections ▪ Report problems and take corrective actions ▪ Perform in-use tests ▪ Maintain configuration management (e.g., maintain software baseline, SWDS, and software change documentation) ▪ Maintain inventory/inventories[4] ▪ Implement supporting quality assurance program controls (e.g., training/qualification and assessments)	▪ Upon SWAU and until the software is retired ▪ In-use testing and assessments as specified on the SWDS (at a minimum, when installed on a different computer or after significant changes)	SU (for use) SO/SRLM (for maintenance)	SOFT-GEN, SOFT-V&V, SOFT-MAINT
2	Retire the software	R	R	R	▪ Prevent unintended or routine use ▪ Terminate software agreements and licenses ▪ Remove from the inventory/inventories as applicable[4] ▪ Where practical or where a future need may exist, retain as a record	When no longer required for routine use or acceptable for use	SO/SRLM	SOFT-GEN, SOFT-MAINT

[1] ML = management level; R = required; Gr = required but graded
[2] SU = software user; SRLM = software responsible line manager (The SU and SRLM may be the same person)
[3] In-use testing is not required for ML-4.
[4] See SOFT-INV for details.
[5] Ref. = ESM Chapter 21 section name

3.1.1 Software Application Documentation

Application documentation for structure, system, and component (SSC) software is a general term that comprises the following main categories.

- **Operations and Maintenance (O&M) Documentation**. Software shall be controlled in accordance with approved procedures and instructions (i.e., O&M instructions). O&M instructions 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 or likely misuse. O&M manual(s) should be listed on the software baseline.
- **Licenses and Registrations.** As applicable, maintain license documentation and other registrations, including maintenance contracts, as appropriate to promote proper software use. As applicable, document licenses on the software baseline and document license and registration maintenance protocols on the SWDS.

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, or other information. As applicable*, indicate event documentation review and retention protocols on the SWDS*

3.1.2 Access Control Requirements

For all ML levels, establish and implement access controls in accordance with P218, *Cyber Security Controls*. Address the security of the computer system and the critical data that may reside in 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. (This requirement may not be applicable in cases such as read-only firmware that lacks user-changeable setpoints.)

3.1.3 Computer System Vulnerability Protections

For all ML levels, establish and implement computer system vulnerability protections as required by PD210, *Cyber Security Program,* and CIO-P100, *Cybersecurity Program Plan.* Base controls on consequence of loss, confidentiality, integrity, and availability. As required, contact an information system security officer to assist with identifying and implementing computer system vulnerability controls. As appropriate, implement the following protections, which may be designed into the program, established as part of the O&M procedures, specified on the SWDS, or a combination of these.

- Access timeouts after a specified time of inactivity.
- Use recovery/contingency protocols, including retention of backups of system data.
- Use software tools to verify pedigree authenticity or detect tampering associated with suspect/counterfeit items (not applicable in all cases, such as firmware).

3.1.4 Problem Reporting and Corrective Action

For all ML levels, follow the problem reporting and corrective action processes in SOFT-GEN.

3.1.5 In-Use Tests

A. When determined necessary by the software owner and SRLM, perform in-use testing of ML-1 through ML-3 software as specified in this section and the SWDS. Perform in-use testing in accordance with test procedures to confirm acceptable performance in the operating environment. Integrate governing SSC work controls.

B. Perform in-use tests in accordance with the applicable test planning and testing requirements of the SOFT-V&V section.

C. Demonstrate required performance over the range of operation of the controlled function or process.

D. Document test procedures and frequency on the SWDS. Specify test frequency using the following methods:

1. Perform testing after the computer program is installed on a different computer.
2. Perform testing when there are changes in the operating system environment when specified by the SRLM.
3. Perform periodic testing where computer program errors, data errors, computer hardware failures, or instrument drift can affect required performance (consider manual and automatic self-check tests).
4. Perform specified testing when required by procedure (e.g., AP-341-801, *Post Modification/Post Maintenance Testing*).
5. Perform testing based on the computer program's risk of failure. For ML-1 and ML-2 software, at a minimum, an annual test is recommended.

3.1.6 Configuration Management

Perform the following tasks:

A. Follow configuration management and interface control processes in SOFT-GEN. Ensure completeness and accuracy in maintaining the software baseline and SWDS.

B. 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.

C. 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.

3.1.7 Inventory

For all ML levels, maintain complete and accurate software inventories in the Form 2033 Graded Software Inventory (GSI) in accordance with P1040, *Software Quality Management.*

3.2 Retire SSC Software

When it is no longer required for routine use or no longer approved for use, the SRLM shall retire software and prevent it from unintended or routine use. Perform the following, as applicable, to retire all ML levels of software (Requirement 21-0602):

A. Terminate software support, including registrations, agreements, and licenses.

B. Remove the software.

C. Update the software record in the GSI in accordance with P1040, *Software Quality Management*.

D. 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.*

E. Where practical 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 this record 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 if it becomes necessary to "roll back" to the software. Retain support software as required to roll back.

F. Cancel or revise affected training courses/plans.

4.0 NON-SSC SOFTWARE

4.1 Use and Maintain

Follow Subsection 3.1 with the following clarifications.

Do not use LANL ML-1, 2, or 3 Non-SSC software unless it is approved for use [i.e., (1) listed as "Completed" including the correct version and (2) current relative to review date on the Form 2033 Graded Software Inventory/GSI, and (3) will be used in accordance with the SWDS], or SWAU is otherwise provided by the SRLM in writing.

For ML-4, acquired software (other than Excel), the Facility Engineering divisions [e.g., Engineering Services (ES), Plutonium Facilities (PFE), and Plutonium Infrastructure Engineering (PIE)] shall also use the GSI (the "Fac Eng Design Record" choice in the left column filters for same).[1] If the software is not listed, contact the SRLM or ES software coordinator.

- Before using the software, the SU must verify that they are authorized to use it by reviewing the SWDS. If the SU is not shown as an authorized user for the software, contact the software owner, SRLM, or ES software coordinator. Do not use the software unless authorization is shown on SWDS or SWAU is otherwise provided by the software owner or SRLM in writing.
- Note, some SSC-related administrative procedures/processes may not apply fully or partially; choose and/or tailor with SRLM concurrence.
- Use the non-SSC change process in SOFT-GEN.
- Maintain application logs for ML-1 and ML-2 non-SSC software that is used for design purposes or in other applications where it may be necessary to trace the impacts of a software error to affected documents (e.g., determining which fire hazards analyses may be impacted by an airflow modeling error). Application logs should record who used the software, the software baseline revision used, and any information necessary to provide traceability (e.g., design change form numbers). Application logs shall be maintained by the software owner. As applicable, provide application log protocols on the SWDS.
- In each calculation where *simple and easily understood* software is used, document the determination of *simple and easily understood* software and review and approve the software (individually verify) with each use as part of the calculation.

4.2 Retire Non-SSC Software

Follow Subsection 3.2 with the following additions:

- As applicable, remove software from the Form 2033/GSI (for Facility Engineering, ref. ESDO-AP-002, *Engineering Service (ES) Software Inventory Instructions)*. This removal includes incomplete software designs.
- Update the GSI software record for safety or non-safety software indicating its retirement.

[1] PFE and PIE may opt out at Division Leader direction.