Section SOFT-INTRO: Chapter Introduction

Rev. 2, 02/11/2026

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*

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, 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.

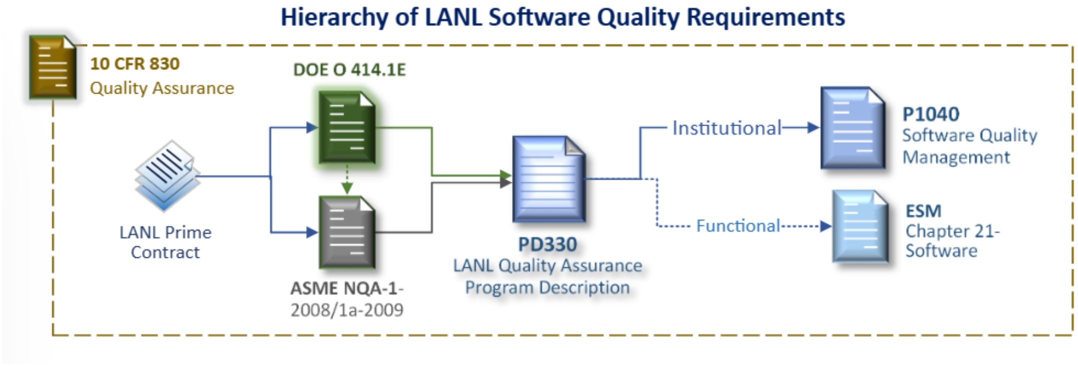


Figure 21.1 Flow Down of Software Quality Requirements

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

[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.

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 (as defined in P1040 [e.g., Att. B]), except for firmware — see ML-based approach below.
  - LTCC software is managed and controlled in accordance with P1040 and IQPA-IQ-FSD-116, *Software Quality Management Plan (SQMP) for LANL Less Than Commercially Controlled (LTCC)* Software *[LANL-internal links, FSD in EDRMS].*

Conditional, Partial Exclusions for SSC Read-Only Embedded Software (aka Firmware):

- ML-4 wherein LANL personnel cannot review the software program; however, the following are required:
  - 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."
  - 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 compensatory measures as ML-3 above.
- ML-1 or ML-2: Same conditions for exclusion as ML-3 and same compensatory measures as for ML-4 above, with these additional requirements:
  - The functional testing is 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.

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

[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.

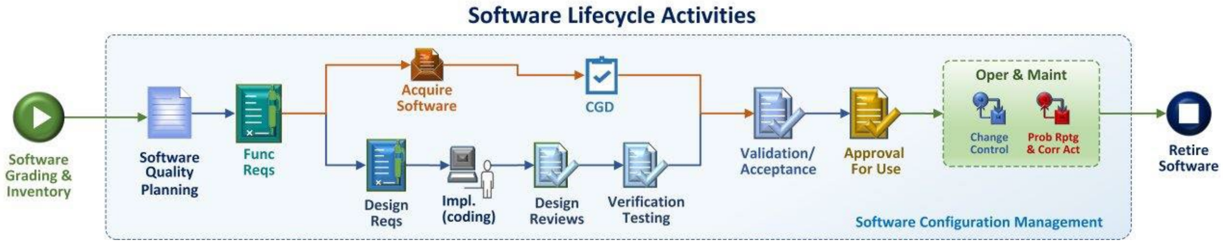


Figure 21.2 Software Life Cycle Activities

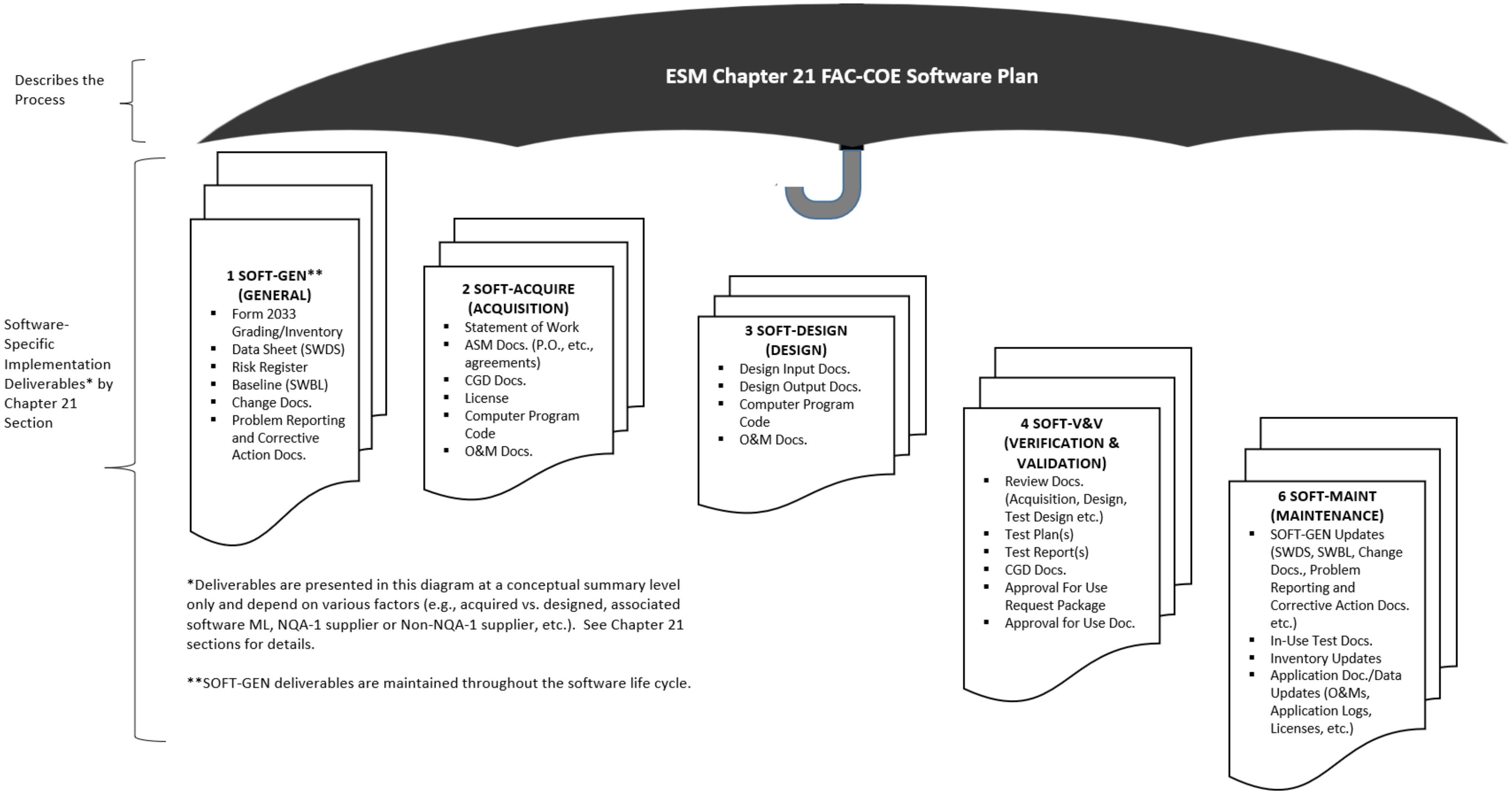
3.1 Chapter Organization

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)


ESM Chapter 21 FAC-COE Software Plan
Describes the Process
Software-Specific Implementation Deliverables* by Chapter 21 Section
1 SOFT-GEN** (GENERAL)
Form 2033 Grading/Inventory
Data Sheet (SWDS)
Risk Register
Baseline (SWBL)
Change Docs.
Problem Reporting and Corrective Action Docs.
2 SOFT-ACQUIRE (ACQUISITION)
Statement of Work
ASM Docs. (P.O., etc., agreements)
CGD Docs.
License
Computer Program Code
O&M Docs.
3 SOFT-DESIGN (DESIGN)
Design Input Docs.
Design Output Docs.
Computer Program Code
O&M Docs.
4 SOFT-V&V (VERIFICATION & VALIDATION)
Review Docs. (Acquisition, Design, Test Design etc.)
Test Plan(s)
Test Report(s)
CGD Docs.
Approval For Use Request Package
Approval for Use Doc.
6 SOFT-MAINT (MAINTENANCE)
SOFT-GEN Updates (SWDS, SWBL, Change Docs., Problem Reporting and Corrective Action Docs. etc.)
In-Use Test Docs.
Inventory Updates
Application Doc./Data Updates (O&Ms, Application Logs, Licenses, etc.)

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

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

Figure 21.3 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).

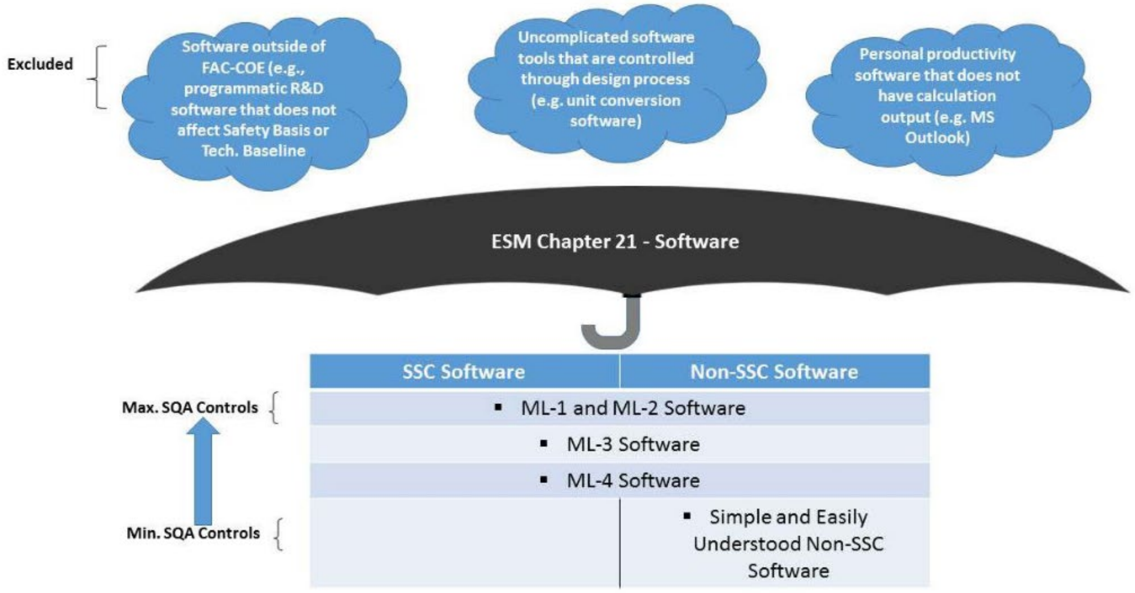

Excluded
Software outside of FAC-COE (e.g., programmatic R&D software that does not affect Safety Basis or Tech. Baseline
Uncomplicated software tools that are controlled through design process (e.g. unit conversion software)
Personal productivity software that does not have calculation output (e.g. MS Outlook)
ESM Chapter 21 - Software
SSC Software
Non-SSC Software
Max. SQA Controls
▪ ML-1 and ML-2 Software
▪ ML-3 Software
▪ ML-4 Software
Min. SQA Controls
▪ Simple and Easily Understood Non-SSC Software

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
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 Inventory*

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