

SANDIA REPORT

SAND 2008-5517

Unlimited Release

Printed January 2009

Sandia National Laboratories Advanced Simulation and Computing (ASC) Software Quality Plan

ASC Software Quality Engineering Practices

Version 3.0

Molly Minana, Jennifer Turgeon, Martin Pilch, and Patricia Hackney

Prepared by

Sandia National Laboratories

Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.osti.gov/bridge>

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: <http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online>



SAND 2008-5517
Unlimited Release
Printed January 2009

SANDIA NATIONAL LABORATORIES ADVANCED SIMULATION AND COMPUTING (ASC) SOFTWARE QUALITY PLAN

ASC SOFTWARE QUALITY ENGINEERING PRACTICES

VERSION 3.0

Molly Minana and Jennifer Turgeon: Software Engineering and Qualification Environments
Martin Pilch: QMU & Management Support
Patricia Hackney: Software Engineering and Qualification Environments

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0826

Abstract

The purpose of the Sandia National Laboratories (SNL) Advanced Simulation and Computing (ASC) Software Quality Plan is to clearly identify the practices that are the basis for continually improving the quality of ASC software products. This plan defines the SNL ASC Program software quality engineering practices and provides a mapping of these practices to the SNL Corporate Process Requirement (CPR)001.3.6: "Corporate Software Engineering Excellence." This plan also identifies ASC management's and the software project teams' responsibilities in implementing the software quality practices and in assessing progress towards achieving their software quality goals.

ACKNOWLEDGEMENTS

The authors of Version 3.0 would like to recognize the authors of the original document (Version 1.0), which remains the foundation for this quality plan: Edward A. Boucheron, Richard R. Drake, H. Carter Edwards, Christi A. Forsythe, Patricia Hackney, Molly Minana, Robert Heaphy, Ann L. Hodges, Constantine J. Pavlakos, Judy E. Sturtevant, Joseph R. Schofield, and Michael Williamson.

Additionally, the authors would like to thank the following individuals for their valuable reviews, comments, and assistance in preparing Version 3.0 of the SNL ASC Software Quality Plan: Judy A. Jones, Timothy G. Trucano, Martin J. Crawford, Joseph R. Schofield, Anthony A. Giunta, Michael W. Glass, Richard R. Drake, and David E. Percy.

Revision History

Date	Revision
10/2008	Publish Version 3.0 (SAND 2008-5517)
2/19/08-8/08/08	Consolidate Version 2.0 Parts 1 and 2 into one document (Version 3.0) that broadens the scope of the ASC SQE practices to fully achieve CPR001.3.6 and maps all practices to the revised CPR.
11/07	Remove 2005 Foreword
10/31/07	Interim Review – includes FY07 review and edits
9/06	Publish Version 2.0 (SAND 2006-5998)
01/05	Publish Version 1.0 (SAND 2004-6602)

TABLE OF CONTENTS

Executive Summary	1
Commitment	1
1.0 Introduction	9
1.1 Drivers and Standards	9
1.2 Version 3.0 Value-Added Summary	11
1.3 Quality Definition and Goals	11
1.4 Document Summary.....	12
2.0 SNL ASC Program SQE Overview	13
2.1 ASC Quality Management Council.....	13
2.2 Management Roles and Responsibilities.....	13
2.3 Stakeholder Expectations	14
2.4 Graded Approach to Levels of Formality.....	14
2.5 ASC Appraisal Methodology and Target Ratings.....	15
3.0 ASC SQE Plan Overview	17
3.1 CPR001.3.6 Components and Impact	17
3.2 Software Quality Plan Components	18
3.3 Software Quality Plan Format and Conventions	21
4.0 ASC SQE Guidelines	23
4.1 Project Management SQE Category.....	23
4.1.1 Integrated Teaming Process Area.....	23
4.1.2 Graded Level of Formality Process Area	24
4.1.3 Measurement and Analysis Process Area.....	25
4.1.4 Requirements Development and Management Process Area	27
4.1.5 Risk Management Process Area.....	28
4.1.6 Project Planning and Oversight Process Area	29
4.2 Software Engineering SQE Category.....	30
4.2.1 Technical Solution Process Area.....	31
4.2.2 Configuration Management Process Area	32
4.2.3 Product Integration Process Area	34
4.2.4 Deployment and Lifecycle Support Process Area	35
4.3 Software Verification SQE Category	36
4.3.1 Software Verification Process Area.....	36
4.4 Training Support Category.....	38
4.4.1 Training Process Area	38
4.5 Practices and Generated Artifacts Summary Table.....	40
5.0 Strategy for Conformance to SQE Practices	43
References	45
Glossary and Acronyms	49
Glossary.....	49
Acronyms	53
Appendix A. Level of Formality Tables: Target Ratings and Rules of Thumb	55
Appendix B. Mapping to Corporate and Federal SQE Requirements	59
Appendix C. Mapping to Version 2.0 of the Software Quality Plan	63
Distribution	65

LIST OF FIGURES

Figure 1. Relationships of Drivers, Software Quality Plan, and Project Implementation	10
Figure 2. Software Quality Plan Organization Relationships	18

LIST OF TABLES

Table 1. CPR001.3.6 Process Areas and Associated Dimensions	17
Table 2. Software Quality Plan Categories, Process Areas, and Practices	20
Table 3. Practices and Generated Artifacts.....	40
Table 4. AQMC Target Rating Level Expectations Based upon Determined LOF.....	56
Table 5. ASC SQE Appraisal Rating Scale	57
Table 6. Rules of Thumb for LOF	58
Table 7. Mapping to CPR001.3.6 Process	59
Table 8. Mapping to Previous Version of the Software Quality Plan.....	63

EXECUTIVE SUMMARY

The purpose of the Sandia National Laboratories (SNL) Advanced Simulation and Computing (ASC) Software Quality Plan is to clearly identify the practices that are the basis for continually improving the quality of ASC software products. Quality is defined in the U.S. Department of Energy/National Nuclear Security Agency (DOE/NNSA) Quality Criteria, Revision 10 (QC-1) as "conformance to customer requirements and expectations." This quality plan defines the SNL ASC Program software quality engineering (SQE) practices and provides a mapping of these practices to the SNL Corporate Process Requirement (CPR) 001.3.6: "Corporate Software Engineering Excellence." This plan also identifies ASC management's and the software project teams' responsibilities in implementing the software quality practices and in assessing progress towards achieving their software quality goals.

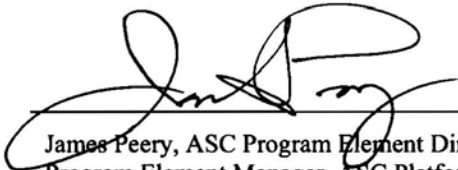
This SNL ASC Software Quality Plan establishes the signatories' commitments to improving software products by applying cost-effective SQE practices. This plan enumerates the SQE practices that comprise the development of SNL ASC's software products and explains the project teams' opportunities for tailoring and implementing the practices.

COMMITMENT

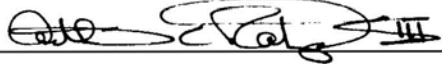
The SNL ASC Program Elements that develop and/or deploy software shall follow the practices, processes, and activities outlined in the SNL ASC Software Quality Plan. Our purpose is to produce quality software products that satisfy our customers' requirements and expectations and provide tangible evidence demonstrating high confidence in ASC software projects at SNL. An additional intent of the SNL ASC Software Quality Plan is to foster organizational consistency by defining common SQE practices and by facilitating the use of cost-effective, common tools and processes where feasible.

This document shall be modified if SQE requirements change.

Approved By


James Peery, ASC Program Element Director and
Program Element Manager, ASC Platforms and Environment

10/20/08
Date


Arthur Ratzel, ASC Program Element Director and
Program Element Manager, Codes and Applications

10/31/08
Date

This page intentionally left blank.

1.0 INTRODUCTION

The National Nuclear Security Agency (NNSA) oversees the Stockpile Stewardship Program (SSP) to provide and ensure confidence in the safety, performance, and reliability of the United States' nuclear stockpile in the absence of underground testing. To this end, NNSA enabled the Accelerated Strategic Computing Initiative (ASCI) to support the SSP in transitioning from using primarily test-based methods to using more computational simulation-based methods. The original ASCI was renamed the Advanced Simulation and Computing (ASC) Program, which fundamentally relies on coordination among the three nuclear weapons laboratories: Lawrence Livermore National Laboratory (LLNL), Los Alamos National Laboratory (LANL), and Sandia National Laboratories (SNL).

This SNL ASC Software Quality Plan (Software Quality Plan) provides the background, high-level information, and overall software quality engineering (SQE) practices that the SNL ASC software projects are required to address and use. This version of the Software Quality Plan replaces Version 1.0 (SAND 2004-6602 published in January 2005)²³ and Version 2.0 (SAND 2006-5998 published in September 2006)² of the document.

Note: Hereafter, the use of the term “ASC” refers to the “SNL ASC” unless otherwise noted.

The Software Quality Plan is a consolidation of the

- previously separate efforts by the ASC Applications and the Simulation and Computer Science/Ongoing Computing (S&CS/OC) programs³⁻⁷,
- feedback from assessments, and
- desire of SNL ASC management to address elements in other quality frameworks (for example, Capability Maturity Model Integration®¹ [CMMI], Institute of Electrical and Electronics Engineers [IEEE], and International Organization for Standardization [ISO 9000]⁸⁻¹⁰).

This Software Quality Plan establishes one plan for all SNL ASC software projects.

1.1 Drivers and Standards

The ASC Program operates under a complex framework of government, industry, tri-laboratories, and SNL corporate policies, directives, and standards. However, the governing document for ASC SQE is the SNL Corporate Process Requirement (CPR) directive, CPR001.3.6: “Corporate Software Engineering Excellence” (CPR001.3.6)¹¹. The CPR001.3.6 incorporates requirements from other key drivers, including

- SNL CPR001.3.2: “Corporate Quality Assurance Program” (CPR001.3.2),¹²
- U.S. Department of Energy (DOE)/National Nuclear Security Agency (NNSA) Quality Criteria (QC-1),¹³ and
- DOE Order 414.1C (DOE 414.1C)¹⁴.

The ASC Program meets the requirements of all of these drivers by meeting the requirements of CPR001.3.6. This Software Quality Plan addresses compliance with CPR001.3.6 software quality guidelines and requirements and is intended for a broad audience. In addition, the ASC Program meets the requirements of *ASCI Software Quality Engineering Goals, Principles, and Guidelines*

[®] CMMI is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

(GP&G)¹⁵. Figure 1 illustrates the relationships of the Software Quality Plan, drivers for this plan, and expected project implementations.

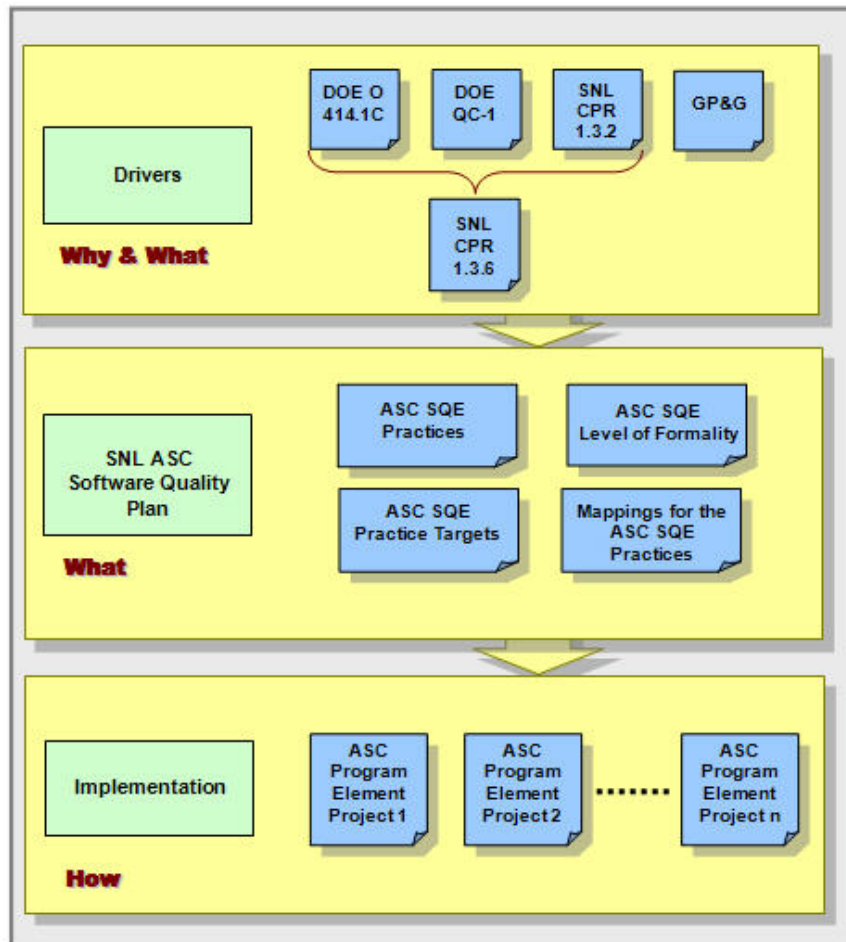


Figure 1. Relationships of Drivers, Software Quality Plan, and Project Implementation

Specifically, the ASC Program addresses CPR001.3.6 directions to

- document a software quality plan that manages and realizes the customers’ requirements,
- use a graded level of formality (LOF) that is determined by a risk-based approach,
- document and implement the process areas and associated dimensions identified in the CPR to the degree appropriate for the risk level determined,
- provide quality records containing evidence that the documented practices are being performed,
- report quality assurance issues, and
- conduct self-assessments and independent appraisals.

The mapping of the Software Quality Plan practices to CPR001.3.6 is provided in Appendix B. This plan shall be reviewed annually under the oversight of the ASC Quality Management Council (AQMC) to consider revisions, including those required to incorporate or otherwise address changes in the governing standards.

1.2 Version 3.0 Value-Added Summary

Previously separate ASC SQE efforts were consolidated into the original Version 1.0 of the Software Quality Plan, which was published in January 2005. The original plan's organization remained essentially the same through Version 2.0, which was published in September 2006. However, CPR001.3.6, the governing document for the Software Quality Plan, was revised substantially in late 2007. As a result of these revisions, ASC management initiated a review and analysis of the Software Quality Plan to ensure that all revised CPR directives were fully addressed. The restructuring of the Software Quality Plan focused primarily on

- broadening the scope of ASC SQE practices to fully achieve CPR001.3.6 directives, and
- mapping all practices to the revised CPR .

The following ground rules were used in this restructuring:

- Create no new practices except as necessary to fill gaps in the Software Quality Plan resulting from CPR001.3.6 revisions.
- Include target ratings (Section 2.5 and Appendix A) for ASC SQE practices.
- Rewrite only those sections required to fill identified gaps to result in minimal new content.
- Focus reviews on new content only. The reviews shall be conducted by Program Element Managers (PEMs) or their designees, resulting in a single point of contact for each program element.

The intent is for this Software Quality Plan to address all guidelines and requirements from ASC SQE drivers and, thereby, eliminate confusion in terminology and scope from multiple directives and reduce the duplication of SQE tasks for both ASC management and projects. Those ASC projects that use a risk-based determined LOF to tailor and implement the 30 ASC SQE practices contained in this Software Quality Plan will meet CPR001.3.6 directives.

1.3 Quality Definition and Goals

The purpose of this document is to describe SQE practices that lead to a high level of confidence in ASC software products and projects at SNL. The intent of the practices stated herein is to promote quality of software products and projects.

The authors of this plan studied multiple sources for defining quality and a common theme surfaced: not all requirements are explicitly stated; however, all implied as well as explicit needs shall be met. Expectations are often defined as customer needs that have not been explicitly stated as requirements. Considering this theme, plus the close traceability between the key drivers (i.e., CPR001.3.6, GP&G), the definition from the QC-1 became the basis of the Software Quality Plan:

Quality - Conformance to customer requirements and expectations

The following quality goals have been identified for this Software Quality Plan:

- Provide guidance for SQE practices to
 - satisfy the stated and implied needs, budget, and schedules of the customer;
 - be effective and cost efficient; and
 - provide a common SQE foundation for ASC projects.
- Enable continual quality improvement of ASC software products, software operation and support activities, and software development activities.

- Satisfy requirements specified in CPR001.3.6, GP&G, and QC-1 to the extent practicable within the scope of this document.

1.4 Document Summary

The following summarizes the document contents:

- Section 1: Introduces the Software Quality Plan, discusses the drivers and standards, briefly summarizes the value added by Version 3.0, and identifies the goals for the document.
- Section 2: Provides a brief ASC SQE overview.
- Section 3: Discusses organizational SQE concepts, pertinent terms, and conventions used in the document.
- Section 4: Identifies the 12 process areas and 30 ASC SQE practices.
- Section 5: Provides the appraisal strategy for conformance to the ASC SQE practices.
- References: Provides information concerning the references and HTML links when available.
- Glossary and Acronyms: Provides definitions used in this document for primary terms and a list of acronyms.
- Appendices: Provide an LOF target rating table and rules of thumb table (Appendix A) and mapping charts (Appendix B and C). For additional suggestions and support for ASC SQE practice implementation, please visit the ASC SQE website¹⁶. This website provides implementation support, including example “best practices,” recommended tools, and templates for implementing practices.

2.0 SNL ASC PROGRAM SQE OVERVIEW

The following are the required general steps for SQE within the ASC Program:

- Document and establish the Software Quality Plan as the software quality standard for the ASC Program.
- Require all ASC projects to implement the practices in the Software Quality Plan. Note: Tailoring of the practices by a project to meet unique characteristics of that project, while retaining the implementation rigor necessary to fulfill its LOF, is allowed if approved and documented.
- Conduct self-assessments, independent appraisals, and external audits to evaluate the implementation of the Software Quality Plan practices by the ASC projects.
- Review and revise the Software Quality Plan as a part of process improvement.

SQE responsibilities, components, and tools that provide the basis for implementing these steps are examined below.

2.1 ASC Quality Management Council

The AQMC is an oversight group responsible for SQE across the ASC Program. ASC subject matter experts are often consulted and may attend AQMC meetings to provide advice. The AQMC responsibilities include, but are not restricted to,

- setting policy and developing strategy for implementing quality systems for all ASC software projects;
- sponsoring and promoting the Software Quality Plan and quality initiatives;
- ensuring that the Software Quality Plan provides a framework for defining and reviewing quality objectives;
- ensuring the Software Quality Plan is communicated and understood by the community;
- reviewing and revising quality documents on an annual basis to ensure currency;
- authorizing modifications to policies and strategies;
- reviewing and assessing quality initiatives in the ASC Program;
- reviewing the results of independent and external appraisals; and
- convening working groups to support development of policies and strategies.

Note: The charter for the AQMC may be found at the ASC SQE website¹⁶. The charter identifies membership, council goals, specific responsibilities, and reporting requirements.

2.2 Management Roles and Responsibilities

The implementation of the SQE practices described in this Software Quality Plan is the joint responsibility of SNL line management and project teams that receive ASC funds. Management support and advocacy of software quality are required for the successful implementation of this Software Quality Plan. ASC line management, which may include several levels of managers, has oversight or other direct responsibilities for ASC-funded software projects. ASC line management ensures consistent and cost-effective implementation of the AQMC's policies and strategies and is responsible for

- directing and ensuring project team implementation of this Software Quality Plan that balances risk, quality, cost, and schedule;
- maintaining the Software Quality Plan;
- approving and tracking the LOF established for projects under its direction; and
- monitoring, improving, and documenting achievement of the requirements of the Software Quality Plan including recording and analyzing quality issues to avoid recurrence.

ASC Program management is responsible for funding the implementation of SQE, including the following activities:

- sponsoring and determining the scope, goals, and procedures of independent SQE appraisals of software projects;
- identifying organizational and stakeholder training needs and providing necessary training opportunities that map to these organizational needs; and
- communicating best software quality practices across the ASC software projects.

2.3 Stakeholder Expectations

Stakeholders are expected to provide guidance, concur with the Software Quality Plan, and participate in the implementation details. A stakeholder is an individual or a group of individuals, internal and external to SNL, who is affected by or is in some way accountable for the outcome or an undertaking in a project. Stakeholders may include project members, suppliers, customers and end users. However, stakeholders need not be directly accountable to the ASC Program; therefore, the Software Quality Plan practices cannot be stated for stakeholders. Project expectations of the stakeholder include

- providing guidance and concurrence with the Software Quality Plan;
- identifying, clarifying, and prioritizing their product expectations and requirements;
- negotiating acceptance criteria, schedule, and intended use of software applications;
- participating in appropriate reviews; and
- identifying customer support expectations and requirements for the installation, operation, and training of the product.

2.4 Graded Approach to Levels of Formality

Each ASC project is expected to determine its LOF, as defined in Section 4.1.2, PR2, of this document. The LOF is determined using a risk-based assessment that analyzes both the consequence of software (product) failure and the likelihood of software (product) failure when the software is used as intended. Based upon a project's LOF, a "graded approach" shall be used to implement the 30 ASC SQE practices. Thus, high LOF projects shall usually implement practices at a more detailed level than medium LOF projects. Low LOF projects shall be accountable for only 10 of the 30 ASC SQE practices (see Table 4, Appendix A), although they may choose to implement more.

The AQMC, with recommendations from SQE subject-matter experts, has identified default target ratings (see Appendix A) that shall be used as a baseline for implementing the 30 practices. Projects shall tailor these target levels and/or exclude certain practices from implementation with written approval of the PEM. Tailoring and approvals for tailoring shall be documented and maintained by the project.

For additional suggestions and support for ASC SQE practice implementation, please visit the ASC SQE website¹⁶. This website provides implementation support, including example “best practices,” recommended tools, and templates for implementing practices.

2.5 ASC Appraisal Methodology and Target Ratings

The ASC Program uses self-assessments, independent appraisals, and external audits to help evaluate the degree to which the ASC projects have implemented the SQE practices, identify best practices being used by the ASC projects, and scope process improvement activities.

The ASC Program uses a formal appraisal methodology that is documented in SAND 2008-0403: *Sandia National Laboratories Advanced Simulation and Computing (ASC): Appraisal Method for the Implementation of the ASC Software Quality Engineering Practices*¹⁶. This method is used in assigning ratings from a scale of 0 to 5 to each practice. Ratings signify the level of implementation of practices by a project team based upon review of objective evidence by an independent appraiser. Appraisal ratings are intended to support planning for ongoing improvements by each project team and across the ASC Program. Section 5.0 and the ASC SQE Appraisal Instrument on the ASC SQE website¹⁶ provide more details about the strategy for project conformance to the ASC SQE practices.

Table 4, Appendix A, contains target ratings set by the AQMC to facilitate ongoing improvements. These ratings are based on priorities of the Program Office with recommendations from subject matter experts and baseline results gathered during the ASC Program-wide appraisals conducted from 2007 through 2008. Target ratings are identified for each practice and organized by high, medium, and low LOF. Target ratings are used to set practice implementation goals specific to a project team’s designated LOF. Once a project has determined its LOF, the target ratings enable the project to conduct a gap analysis between its current state of practice implementation and the AQMC’s identified targets.

This page intentionally left blank.

3.0 ASC SQE PLAN OVERVIEW

3.1 CPR001.3.6 Components and Impact

The major driver for restructuring the Software Quality Plan is revision of the “process areas” and “associated dimensions” in CPR 001.3.6, which was re-released in October 2007. Table 1 provides a listing of these CPR requirements.

Table 1. CPR001.3.6 Process Areas and Associated Dimensions

Process Area	Associated Dimensions
Project Planning & Oversight	<ul style="list-style-type: none">• Identified and involved stakeholders• Ongoing process monitoring and control• Collected improvement information• Objective evaluations• Quantitative objectives defined for processes• Stable subprocess performance• Training• Problem reporting and corrective action
Risk Management	
Requirements Development & Management	
Technical Solution	
Verification & Validation	
Deployment & Life Cycle Support	
Configuration Management	
Measurement & Analysis	
Integrated Product & Teaming	

All of the CPR001.3.6 “process areas” were addressed in the “practice areas” of Version 2.0 of the Software Quality Plan, but the names and scopes of the practice areas were not always the same as the process areas in the CPR. The following general changes to the Software Quality Plan are meant to remedy possible confusion and to ensure the CPR001.3.6 process areas and associated dimensions are fully addressed:

- The names of the Software Quality Plan “practice areas” were changed to the names of the CPR “process areas” with the following two exceptions:
 - The CPR process area “Integrated Product & Teaming” is documented in two process areas in the Software Quality Plan (i.e., Sections 4.1.1, Integrated Teaming, and 4.2.3, Product Integration).
 - The CPR “Verification & Validation” process area is the “Software Verification” process area in the Software Quality Plan.
- The term “practice area” was replaced with term “process area” throughout the Software Quality Plan.
- The order of the process areas is not the same in both documents. This was to limit the restructuring of Version 3.0 to significant value-added changes. The process areas are grouped into “categories” in the Software Quality Plan to more easily identify and manage related processes.
- A complete mapping of the Software Quality Plan to the CPR00.1.3.6 is provided in Table 7, Appendix B.
- The “associated dimensions” in the CPR, with the exception of training, are not separate requirements in the Software Quality Plan but are, instead, embedded in the wording and scope of key defining practices that are to be globally implemented across all 30 practices.

3.2 Software Quality Plan Components

The Software Quality Plan is organized into three SQE categories (i.e., Sections 4.1, Project Management; 4.2, Software Engineering; and 4.3, Software Verification) and one support category (i.e., Section 4.4, Training). Each category is subdivided into one or more related process areas (previously known as practice areas in Versions 1.0 and 2.0). For example, the Software Engineering category contains four related process areas (i.e., Sections 4.2.1, Technical Solution; 4.2.2, Configuration Management; 4.2.3, Product Integration; and 4.2.4, Deployment and Lifecycle Support). Each of the 12 process areas contains groupings of related SQE practices. Figure 2 depicts the high-level interrelationships among the categories, process areas, practices, and embedded dimensions in key practices.

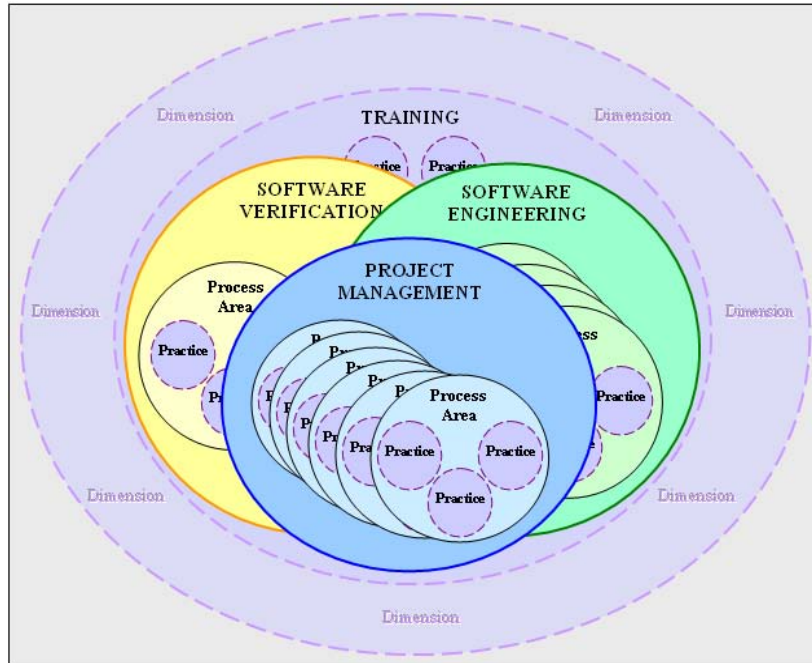


Figure 2. Software Quality Plan Organization Relationships

The 30 practices identify the core concepts that guide SQE implementation. For example, the Technical Solution process area contains four practices (i.e., PR13, PR14, PR15, and PR16). The 30 ASC SQE practices presented in Section 4.0 include detailed information about what shall be addressed to meet a particular SQE concept. Practices contain actions that shall be documented by project teams. Generally, this documentation is provided via artifacts that demonstrate implementation of a practice. During SQE appraisals, these artifacts, individually and collectively, become “objective evidence” that is used to support practice implementation.

CPR001.3.6 directs that eight dimensions (i.e., SQE principles that are intended to be applied to all practices not just within a specific practice) be associated appropriately with the process areas. Many of these dimensions were already embedded in the wording and scope of existing practices in Versions 1.0 and 2.0 of the Software Quality Plan. In fact, the dimension for “training” was determined to be of such high value in these previous versions that it was established as a separate “support category” for the three SQE categories. The organization of the Training Support category was not changed and the other dimensions from the CPR 001.3.6 were addressed and defined by broadening the scope of a few key practices (see Table 7, Appendix B).

A project team shall determine the level of detail to which each practice shall be documented and to what extent each dimension shall be addressed by each practice (see PR2 in Section 4.1.2 and PR3 in Section 4.1.3). Both the tailoring and the rationale for tailoring decisions shall be discussed and documented by the project team and approved by appropriate levels of management.

The organization into process areas grouped by categories helps separate the practices into sets of manageable activities; it is the practices that the ASC projects are responsible for implementing to meet the intent of the process areas and, thereby, the Software Quality Plan. Additionally, dimensions embedded within the key practices shall be extended globally to all practices and all process areas. Table 2 summarizes the Software Quality Plan categories, process areas, and practices, which will be discussed in detail in Section 4.0.

Table 2. Software Quality Plan Categories, Process Areas, and Practices

SQE Categories/Process Areas/Practices
Project Management SQE Category
1. Integrated Teaming
PR1. Document and maintain a strategic plan.
2. Graded Level of Formality
PR2. Perform a risk-based assessment, determine level of formality and applicable practices, and obtain approvals.
3. Measurement and Analysis
PR3. Document, monitor, and control lifecycle processes and their interdependencies, and obtain approvals.
PR4. Define, collect, and monitor appropriate process metrics.
PR5. Periodically evaluate quality issues and implement process improvements.
4. Requirements Development and Management
PR6. Identify stakeholders and other requirements sources.
PR7. Gather and manage stakeholders' expectations, requirements, and constraints.
PR8. Derive, negotiate, manage, and trace requirements.
5. Risk Management
PR9. Identify and analyze risk events.
PR10. Define, monitor, and implement the risk response.
6. Project Planning and Oversight
PR11. Create and manage the project plan.
PR12. Track project performance versus project plan and implement needed (i.e., corrective) actions.
Software Engineering SQE Category
7. Technical Solution
PR13. Communicate and review design.
PR14. Create required software and product documentation.
PR15. Identify and track third party software products and follow applicable agreements.
PR16. Identify, accept ownership, and manage assimilation of other software products.
8. Configuration Management
PR17. Perform version control of identified software product artifacts.
PR18. Record and track issues associated with the software product.
PR19. Ensure backup and disaster recovery of software product artifacts.
9. Product Integration
PR20. Plan and generate the release package.
PR21. Certify that the software product (code and its related artifacts) is ready for release and distribution.
10. Deployment and Lifecycle Support
PR22. Distribute release to customers.
PR23. Define and implement a customer support plan.
PR24. Implement the training identified in the customer support plan.
PR25. Evaluate customer feedback to determine customer satisfaction.
Software Verification SQE Category
11. Software Verification
PR26. Develop and maintain a software verification plan.
PR27. Conduct tests to demonstrate that acceptance criteria are met and to ensure that previously tested capabilities continue to perform as expected.
PR28. Conduct independent technical reviews to evaluate adequacy with respect to requirements.
Training Support Category
12. Training
PR29. Determine project team training needed to fulfill assigned roles and responsibilities.
PR30. Track training undertaken by project team.

3.3 Software Quality Plan Format and Conventions

Under each of the SQE categories discussed in Section 4.0, the overall scope is summarized for the category and then delineated in one or more process areas. Each process area contains an overview description with the expectations of ASC management, statements of the numbered practices, supplemental practice guidance if applicable, required artifacts that demonstrate implementation, and example inputs.

Overview Description: The overview description provides a high-level discussion of particular practices that are included in an area. The overview also provides additional elaboration that is intended to guide the practitioner in implementing the practices described. The overview of one section may reference the overview or details of another related section.

Practices: ASC SQE practices are software development and deployment activities. Each practice describes the activities and elements that a project team shall address in tailoring and implementing the practice for their specific project. Each practice is uniquely numbered in the format “**PRx**.”

Supplemental Practice Guidance: Supplemental practice guidance is provided for some practices to institute a common ASC Program standard of response by the project teams to a particular practice. For example, supplemental practice guidance regarding a graded approach for determining an LOF is provided for PR2: “Perform a risk-based assessment...” The format “**SPGx**,” where the “x” references the number of the associated PR, is used for supplemental practice guidance. For example, SPG2 identifies the supplemental practice guidance associated with PR2. If multiple supplemental practice guidance entries apply to one PR, the format will be further appended with alpha characters, such as “SPG2a,” SPG2b,” and so forth.

Artifacts: An artifact is a required deliverable or work product that is generated as a practice is exercised. Each listed artifact is an example of an output created or modified by the given practice. All required software product artifacts identified by the project team are to be version controlled and change managed, which is described in Section 4.2.2, Configuration Management. Each artifact is uniquely numbered in the format “**ARx**.” Text that is in parentheses following the artifact name is intended to help clarify or explain the artifact. Text that is in brackets following the artifact name is intended to identify elements that might be included in the artifact. Each artifact is followed by a parenthetical expression indicating the practice where the artifact is originated.

Table 3, Section 4.5, provides a joint listing of the SQE practices and the required artifacts generated by each practice.

Example Inputs: The inputs suggested in Section 4.0 are examples of existing resources, information and/or artifacts external to a practice that may support performance of that practice. For an example of an external resource, note that Section 4.1.5, Risk Management, shows the suggested input as a “list of subject matter experts knowledgeable about potential risk events.” For an example of an artifact from another practice, note that most of the suggested inputs in Section 4.2.4, Deployment and Lifecycle Support, are artifacts from other process areas. A suggested input that is a resource or information external to the practice is identified by a bullet (*) and one that is an artifact from another process area is identified by that artifact’s number (ARx). Each suggested input is followed by a parenthetical expression indicating the associated practice that may use the example input(s).

This page intentionally left blank.

4.0 ASC SQE GUIDELINES

4.1 Project Management SQE Category

Project Management is the systematic approach for balancing the project work to be done, resources required, methods used, procedures to be followed, schedules to be met, and the way that the project is organized. This section begins with 4.1.1, Integrated Teaming, as a first step in addressing project management. The second step involves a risk-based assessment presented in 4.1.2, Graded Level of Formality. Thereafter, process areas address specific project management activities in 4.1.3, Measurement and Analysis; 4.1.4, Requirements Development and Management; 4.1.5, Risk Management; and 4.1.6, Project Planning and Oversight.

4.1.1 Integrated Teaming Process Area

Overview Description

Within the Integrated Teaming process area, an organization defines a project and its mission, management and project team member responsibilities and authorities, users and customers, and interrelationships with stakeholders and other projects. The project's mission is one basis for the selection of appropriate practices. For example, a research project may not need all practices used by a team that develops software for a production environment. Integrated teaming presents an opportunity for organizationally related projects to share common practices, procedures, processes, tools, training, and documentation. Large projects and frameworks may form their own organizational contexts that allow subteams to work at their own appropriate LOFs and with their own appropriate practices within the project.

The defined mission of the project implies the intended use of products over which the project has responsibility. A project mission may be exploratory (for example, to develop knowledge or skills) and is not intended to produce a deliverable product. The mission may be to support a pre-existing product that is delivered to customers (for example, a legacy code). A project's mission may cover the full lifecycle of a product from inception through delivery. A single project may have multiple missions; for example, a software product may contain mature features (support mission), features under development (development mission) and research features that are not yet intended for customers (research mission).

A project's strategic plan defines the project team members' functional roles and responsibilities, management responsibilities and authority, users and customers, and interrelationships among all stakeholders and other projects. Management includes the AQMC, PEMs, and other line management as appropriate. An organization, a cooperating group of projects, or a framework project may share documentation for shared practices, processes, and tools.

Practices

PR1. Document and maintain a strategic plan.

The mission (or scope) of the project is clearly defined, documented, and updated when the mission changes. Management responsibilities and authorities for the project are clearly defined, documented, and updated. The initial identification of project stakeholders is addressed in the strategic plan. Commitments for changes to mission and organizational context are only negotiated by authorized personnel with appropriate technical inputs. This practice includes establishing authorities and identifying initial sources of technical inputs. Additionally, the strategic plan defines team operating procedures (such as communication types and frequency, decision-making process, issue-escalation procedures, stakeholder involvement, and interactions

throughout the project life cycle, etc.) to be used by the project. The strategic plan shall be reviewed and updated at least annually by the project team and relevant stakeholders.

Artifacts

AR1. Strategic plan [project’s mission, management, stakeholders, stakeholder roles and responsibilities, team operating procedures] (PR1)

Example Inputs

- Organization representatives (PR1)

4.1.2 Graded Level of Formality Process Area

Overview Description

Each ASC software project applies a risk-based assessment to determine its LOF in implementing applicable practices. This risk-based assessment considers factors such as the intended use of the product, product mission and complexity, budget and schedule pressure, customer and public confidence, safety and security, stability of requirements, and the total funding investment that is expected to be applied to this product over its entire lifecycle. The default is that all project teams shall address all practices with a LOF that is consistent with the consequence and likelihood of software (product) failure, as defined in Supplemental Practice Guidance SPG2a, when used as intended; however, some practices could be deemed “not applicable” with ASC PEM approval. The project performs this risk-based assessment with involvement from management and other stakeholders, and the appropriate PEM reviews and approves resulting LOF and applicable practices. The initial risk-based assessment is performed at the beginning of a project, and reassessments are performed when LOF factors change significantly. A reassessment initiates corrective actions to bring the project’s LOF and applicable practices into achievement of the demanded requirements. Product risks are the main focus of this section. See Section 4.1.5 for project risk management practices.

Practices

PR2. Perform a risk-based assessment, determine level of formality and applicable practices, and obtain approvals.

Perform a risk-based assessment based upon the product’s consequence and likelihood of failure. The “ASC Risk-based Assessment Procedure” (see SPG2a) is a template that provides guidance for performing a risk-based graded approach to SQE practice implementation. ASC recognizes three LOFs (high, medium, and low) and requires all practices to be addressed with increasing rigor for the higher levels. Projects with high formality codes are required to employ more rigor than those with medium formality codes, but both are expected to address all 30 ASC SQE practices. Projects with low formality codes are expected to address a specified subset of the 30 (10 practices; see Table 4, Appendix A), typically with the same rigor as projects with medium formality codes. Each ASC project shall tailor this guidance to better align with their implementation strategy. The tailoring considers the default target ratings (i.e., levels of rigor) associated with each of the levels of LOF (see Table 4, Appendix A). Additionally, a safety software assessment shall be conducted annually by all project teams. (See SPG2b for supplemental practice guidance regarding safety software determination.) Results of this assessment directly affect the consequence of failure for the product and shall affect the overall LOF for a project team.

Decide on the applicable practices and level of formality that will mitigate the risk level. The project team analyzes the intended use of the product as defined in its strategic plan (see PR1) to

guide the determination of which practices to implement. The project team shall tailor its implementation of the 30 ASC SQE practices based upon the results of its risk-based assessment (i.e., its documented consequence level and likelihood of failure). This tailoring shall include a decision both on which practices are applicable and on the LOF to be applied in implementing these selected practices. The AQMC expected LOF target ratings for each practice (see Table 4, Appendix A) shall be considered as a baseline for tailoring. Example “rules of thumb” for LOF issues related to artifacts, reviews, training, and tool usage are provided in Table 6, Appendix A.

A PEM reviews, approves, and tracks the assessed LOF and applicable practices for ASC projects in his/her program element. A project shall request a waiver from the AQMC’s expected target ratings and for any practices it deems not applicable. The waiver requires written approval from the PEM. The LOF documentation and associated waivers are maintained by the project team as part of the project’s evidence base and shall be evaluated annually. Supplemental Practice Guidance SPG2a provides a template for documenting this information.

Supplemental Practice Guidance

SPG2a Risk-based assessment procedure to determine LOF (template): ASC Risk-Based Assessment Procedure¹⁸; available at: <https://wfsprod01.sandia.gov/groups/srn-uscitizens/documents/document/wfs680943.pdf>.

SPG2b ASC safety software guidance: Instructions for Determining Whether An ASC Software Product Should Be Categorized As DOE Order 414.1c Safety Software¹⁹, available at: <https://wfsprod01.sandia.gov/groups/srn-uscitizens/documents/document/wfs710561.pdf>.

Artifacts

AR2 Approved level of formality and approved applicable practices [tailoring and/or waivers] (PR2)

Example Inputs

- Customer and organization process requirements (PR2)

AR1 Strategic plan [project’s mission, management, stakeholders] (PR2)

4.1.3 Measurement and Analysis Process Area

Overview Description

Process implementation typically includes the activities required to plan, define, implement, monitor, measure, and improve all aspects of a product lifecycle from concept to retirement. Examples of lifecycles include iterative, spiral, and concurrent. Various methodologies can be employed to support software lifecycles. Practices are implemented through lifecycle processes that define the activities, interfaces, roles, and responsibilities. (See the Glossary for a definition of “process” as used in this document.)

Process improvement is the continual activity, which uses measurement and analysis, to increase the ability of a process to meet its objectives. Lifecycle processes are evaluated by monitoring, measuring, and analyzing their effectiveness and efficiency with respect to project team and ASC Program objectives. This evaluation is used to investigate alternative improvement solutions and select cost-effective improvements to the processes. An objective for process evaluation and improvement is to anticipate and prevent product errors and nonconformance. Issues, errors, or nonconformance are analyzed to determine if corrective actions are required to improve the processes and prevent recurrence of similar issues. Process improvement changes are reviewed, managed, and documented. See the ASC SQE website¹⁶ for suggested effective metric and non-metric based process improvement techniques and tools.

Self-assessments and independent appraisals of processes shall be conducted to monitor implementation and identify areas for improvement.

These measurement and analysis practices are viewed as a cross-cutting dimension of this Software Quality Plan and shall be addressed across all ASC SQE practices to ensure that quantitative process objectives (target values) are defined, managed, and improved. Measurement and analysis practices shall be treated separately from practices in Section 4.1.6, Project Planning and Oversight, to allow organizations to define common lifecycle processes that may be shared and followed by multiple projects.

Practices

PR3. Document, monitor, and control lifecycle processes and their interdependencies, and obtain approvals.

The project team defines and documents its applicable lifecycle processes by considering the LOF, intended use, project and ASC Program objectives, cost, resource constraints, and compatibility with customers and other projects' activities. Defined lifecycle processes shall include activities, interfaces, roles, and responsibilities. The appropriate stakeholders review and appropriate management approves the documented lifecycle processes. Once documented, processes are monitored against plans for implementing the processes within the project and ASC Program expectations (see target ratings in Table 4, Appendix A). Issues with and deviations from planned processes shall be reviewed with appropriate stakeholders to determine any needed corrective actions to the processes. Corrective actions and issues shall be recorded and monitored to closure (see PR18).

PR4. Define, collect, and monitor appropriate process metrics.

The AQMC set LOF target rating expectations (see Table 4, Appendix A) as quantitative process objectives across the ASC Program and identified three metrics (i.e., achievement ratings from assessments and appraisals for the 30 practices and the resulting gaps between those ratings and the target ratings in Table 4, Appendix A, test coverage, and defect metrics) that are to be collected by all ASC projects. In addition, each project team defines metrics and utilizes customer feedback to aid in the evaluation of process effectiveness and efficiency. Typically a new team identifies only selected metrics that will add immediate value to improving their processes or in the way they approach their lifecycle activities. As the project evolves, the number of metrics collected may increase and expanded evaluations of a particular metric shall be established to address additional areas where improvements are needed based upon desired target values. Expanded evaluations of a particular metric generally involve statistical management of a subprocess to remove any special causes of process variation. Issues identified from monitoring metrics shall be recorded and monitored to closure (see PR18).

PR5. Periodically evaluate quality issues and implement process improvements.

Conduct periodic self-assessments. The project team monitors conditions in order to investigate and prioritize solutions to quality issues. The team is responsible for documenting and implementing improvement solutions. Typically the project team analyzes metrics to aid in this evaluation and shall communicate results with relevant stakeholders. Additionally, project teams conduct self-assessments (see Section 5.0) to identify the state of process implementation within the project team. Issues identified during self-assessments shall be recorded and monitored to closure (see PR18).

Conduct periodic independent appraisals. Project teams shall use the results of independent appraisals to obtain objective evaluations of process implementation to support improvement efforts and obtain credible assurance that processes are implemented as planned. An individual(s)

who is knowledgeable in the relevant subject area and is independent of the project team and of the processes/activities being evaluated or appraised shall conduct independent project team evaluations. These evaluations shall be managed by the ASC Program Office (see Section 5.0). Issues identified during such appraisals shall be recorded and monitored to closure (PR18).

Artifacts

- AR3** Approved project processes (PR3)
- AR4** Process and product metrics [defined metrics, quantitative objectives for process performance, statistically managed measurements for a subprocess] (PR4)
- AR5** Project process improvement actions (PR5)
- AR16** Managed issues [process issues, product quality results (for example, non-conformances), enhancements, defects, inquiries] (PR3, PR4, PR5)
- AR27** Self-assessment and independent appraisal results (PR5)

Example Inputs

- Customer and organization process requirements (PR3)
 - Information on available and planned resources (PR3)
 - Previous self-assessment and independent appraisal results (PR5)
- AR1** Strategic plan [project’s mission, management, stakeholders] (PR3, PR4)
- AR2** Approved level of formality and approved applicable practices [tailoring and/or waivers] (PR3)

4.1.4 Requirements Development and Management Process Area

Overview Description

The purpose of requirements development and management practices is to capture, analyze, develop, validate, track, and control the product requirements and specifications throughout the lifecycle of the product. Product requirements typically span hardware, software, operations, support, documentation, product training, and other aspects. Requirements are based upon project mission, stakeholders’ stated and implied “needs,” and organizational commitments. Although needs are not requirements they are considered along with requirements in order to improve quality. Changes to requirements shall be managed throughout the lifetime of the project.

Requirements are inputs to other process areas. Risk management activities analyze and try to control events that affect the ability to satisfy requirements. Project planning determines whether and when requirements will be implemented. A product release identifies requirements that are newly satisfied in that release. Software verification reviews evaluate whether the product has met the requirements according to specified acceptance criteria. Requirements shall be reviewed and approved by appropriate stakeholders.

Practices

PR6. Identify stakeholders and other requirements sources.

Sources of requirements potentially include stakeholders as well as regulatory, historical, organizational, and computational commitments. The project team communicates with the customers and other stakeholders regarding areas needing support. Stakeholders also include suppliers of products that are to be integrated with the project team’s product. Stakeholder identification and involvement is viewed as a cross-cutting dimension of this Software Quality Plan and shall be addressed across all ASC SQE practices to identify when stakeholder involvement and interactions are necessary. This helps ensure that work is conducted efficiently and the correct individuals are involved in decision making.

PR7. Gather and manage stakeholders' expectations, requirements, and constraints.

Product expectations and requirements are gathered from identified stakeholders, additional commitments, and submitted issues. The gathering activity includes identifying the source, criticality, priority, and acceptance criteria of the needs. There may be needs that are not clear. In these cases the originator shall be contacted for further clarification. These sources may include stockpile drivers, expectations of fitness for intended use, programmatic requirements, physical or functional requirements, modeling or simulation requirements, or issues submitted for a previous version of derived software requirements.

PR8. Derive, negotiate, manage, and trace requirements.

The software project team derives and negotiates software requirements based upon the gathered needs and analysis of technical feasibility and resource availability. Negotiation optimally includes project team and stakeholder approvals of derived requirements and subsequent delivery commitments. Projects shall document the “forward tracing” of general requirements to more detailed requirements and product components (e.g., code, design, tests, etc.) and the “backward tracing” of product components to their general requirements. Ideally, requirements traceability supports analyzing the impact of the change.

Artifacts

- AR6** Product expectations and requirements (PR6, PR8)
- AR7** Software requirements and attributes (PR7, PR8)
- AR8** List of stakeholders and organizational commitments (PR6, PR8)

Example Inputs

- Stakeholder expectations, requirements, and constraints (PR7)
 - Organizational requirements (PR7)
 - Platform requirements and characteristics (PR7)
- AR1** Strategic plan [project's mission, management, stakeholders] (PR7)
 - AR16** Managed issues [enhancements, defects, inquiries] (PR7, PR8)
 - AR19** Customer support plan including training (PR7)

4.1.5 Risk Management Process Area

Overview Description

Risk management is the activity of identifying, addressing, mitigating, and tracking sources of risk before they become threats to successful completion of a project. A risk is a combination of the consequence and likelihood of an event. Risk management spans the lifetime of the project. The number of risks and risk factors is unbounded. Therefore, this process area seeks to identify risks associated with planned work from a variety of sources including organizational risk matrices, customer feedback, audit and appraisal results, technical evaluation of the work, corporate lessons learned, and experience from other projects and activities. Risk management is intended to mitigate consequences and/or likelihood of these identified risk events. Prior to identifying, analyzing, and monitoring risk, it is important to define a workable strategy that shall be used to manage project risk. Such a strategy shall address methods and tools that will be used; identify sources; explain how risks will be organized, categorized, compared, and consolidated; and specify who will monitor risks and how often that monitoring will occur. Monitoring risk events shall be done in conjunction with the practices in Section 4.1.6, Project Planning and Oversight.

Practices

PR9. Identify and analyze risk events.

Significant risk events shall be identified and clearly described. As conditions change, identified risks shall be reviewed and updated in a risk plan. The risk analysis process shall identify key attributes of each risk event such as the consequence, likelihood, group(s) impacted by the risk event, and the organization (i.e., risk owner) responsible for any action associated with the risk event. Risk events are prioritized based on consequences, likelihood, and potentially other factors.

PR10. Define, monitor, and implement the risk response.

A risk response comprises the risk disposition and corrective action(s) for events to be mitigated. Given a prioritized set of risk events, the project determines the risk disposition of the highest priority events. Possible dispositions include actions to “mitigate,” “transfer,” “accept,” and “avoid.” Teams shall plan a response for unanticipated events that threaten the successful completion of the project.

Projects monitor risk by collecting relevant information. The monitoring approach is documented in a risk plan and includes who does monitoring, how often, how information is collected, tools to assist monitoring, etc. If a risk event occurs, the planned corrective actions are implemented, which includes notification of impacted stakeholders.

Artifacts

AR9 Project plan [risks events, risk plan] (PR9)

AR10 Project reviews and needed (corrective) actions: [risk responses] (PR10)

Example Inputs

- List of subject matter experts knowledgeable about potential risk events (PR9)

AR6 Product expectations, requirements, and constraints (PR9)

AR7 Software requirements and attributes (PR9)

AR8 List of stakeholders and organizational commitments (PR9)

AR9 Project plan [risk events, risk plan] (PR10)

AR10 Project reviews and needed (corrective) actions [tracking and oversight responses] (PR9)

AR24 Technical reviews (PR9)

4.1.6 Project Planning and Oversight Process Area

Overview Description

Project planning and oversight guides project implementation while balancing, monitoring, and analyzing project quality, cost (including cost of quality), schedule, and performance. Project planning includes preparing a plan that describes how the project shall be performed and managed. The plan shall include, at a minimum, a statement of work, project constraints and goals, project deliverables, a project timeline, needed acquisitions and purchases, an assessment of required resources, and the availability of the resources. ASC and stakeholder organizations use the project plan to fund, plan, and provide a basis for tracking and oversight. The project plan shall be reviewed and updated throughout the lifetime of the project.

Oversight includes taking corrective actions as necessary. Corrective actions bring projected accomplishments and results back into conformance to the defined requirements. Corrective actions could include adding resources to meet schedules, modifying the schedule, adding project budget, modifying cost criteria, and re-negotiating requirements or acceptance criteria.

Practices

PR11. Create and manage the project plan.

Project plans typically contain a project overview, project tasks, resource information, planning assumptions and constraints, dependencies, budget, schedule, and roles and responsibilities. This practice includes identifying tasks and evaluating feasibility, cost, resource requirements, and both internal and external dependencies of the tasks. The plan shall be reviewed with those responsible for its implementation with the goal of agreed-upon commitment from such parties.

For additional suggestions and support for ASC SQE practice implementation, please visit the ASC SQE website¹⁶. This website provides implementation support, including example “best practices,” recommended tools, and templates for implementing practices.

PR12. Track project performance versus project plan and implement needed (i.e., corrective) actions.

The project team shall determine what project metrics are of interest, then monitor and analyze these metrics by comparing actual performance and progress against what was planned and recorded in the project plan. This monitoring may be performed via automated tools or manually, and shall occur frequently to allow time to identify and analyze any significant variances prior to project impact. For significant variances, the root cause and potential corrective actions shall be determined. This activity shall include discussion with stakeholders and management concerning the severity and impact of the identified variances.

Artifacts

AR9 Project plan [risk events, risk plan, overview, milestones, task list, resource information, roles and responsibility assignments, assumptions, constraints, dependencies, budget, schedule, software configuration management (SCM) plan, etc.] (PR11)

AR10 Project reviews and needed (corrective) actions [risk responses, tracking and oversight responses] (PR12)

Example Inputs

- Information on available resources (PR11)
- AR1** Strategic plan [project’s mission, management, stakeholders] (PR11)
- AR3** Approved project processes (PR12)
- AR4** Process and product metrics (PR12)
- AR6** Product expectations and requirements (PR11)
- AR7** Software requirements and attributes (PR11)
- AR8** List of stakeholders and organizational commitments (PR12)
- AR9** Project plan [risk plan, risk events] (PR12)
- AR19** Customer support plan including training (PR11)

4.2 Software Engineering SQE Category

Software engineering is a systematic approach to the specification, design, development, test, operation, support, and retirement of software. The software engineering process areas include Sections 4.2.1, Technical Solution; 4.2.2, Configuration Management; 4.2.3, Product Integration; and 4.2.4, Deployment and Lifecycle Support.

4.2.1 Technical Solution Process Area

Overview Description

The purpose of the technical solution practices is to generate a correctly working product for the customer; this product is often, but not always, software. Generally, software development processes include design, implementation, and testing of the software products or reuse of existing implementations. Other practices related to software development are covered elsewhere in this document: Sections 4.1.4, Requirements Development and Management; 4.2.2, Configuration Management, which includes version control and issue tracking; and 4.3, Software Verification, for reviews and testing. The lifecycle processes are documented in Section 4.1.3, Measurement and Analysis. The Software Quality Plan prescribes no specific lifecycle or any particular software development methodology.

Projects sometimes rely on the incorporation of third party or other existing software products in order to satisfy needed capabilities without incurring the cost of redeveloping those capabilities. Such software may be a simple library, an integrated set of libraries, compilers and linkers, or even an operating system. Sources of such software may be commercial, open source, other ASC or SNL projects, or research efforts. Third party or other existing software that becomes integrated with the internally developed software requires the implementation of activities such as identifying, tracking, establishing trust in, assimilating, or honoring agreements (for example, protecting intellectual property) in order to propagate the same quality standards throughout the developed product. Note that requirements traceability (see PR8) includes tracing requirements satisfied through the integrated third party or other existing software.

Practices

PR13. Communicate and review design.

Design is the process of defining architecture, components, interfaces, and other characteristics of a system or components. Design activities transform requirements into artifacts that are used for the development of software. Design artifacts capture information and process specifications that document dependencies, information flows, algorithms, the interfaces, and all the components. These activities and artifacts help ensure requirements are implemented and team members have a common understanding of the design. The impact of implementation choices on design is continuously incorporated. Relevant stakeholders shall be informed of issues and included in decisions. Documentation of a design shall support development, product maintenance, tracing of requirements, verification, and end users. Design reviews are an important aspect of software development and shall be conducted before product implementation or modification. Depending upon the software methodology being used by a project team, design artifacts may not be simultaneously available for formal reviews so informal design reviews and design artifacts shall provide the quality necessary for this practice under these circumstances. See Table 4, Appendix A, for suggestions on carrying out LOF for artifacts and reviews.

PR14. Create required software and product documentation.

The project team shall create the required product artifacts (such as code, user documentation, developer's guide, and installation guide) using the documented project processes. Note that testing and review of these products is part of software verification. These artifacts implement the requirements and are reviewed and updated to reflect the "as built" product at each major product release.

PR15. Identify and track third party software products and follow applicable agreements.

A project typically uses third party software products without modification. However, if the project does modify the third party software, those modifications shall be tracked until the supplier incorporates them into the provided third party software. A project may acquire and configuration manage software (for example, public domain software) or may use software “as-is” in the computational environment (for example, a compiler). A third party software product, its source, and the project’s basis for trust in that product shall be identified. Trust could be based simply upon the supplier’s long-standing reputation or confirmation that another trusted project has already established trust in the third party software, or could involve more complex verification efforts. Applicable agreements with a third party software product supplier could include, for example, licenses, protection of intellectual property, or customer support.

PR16. Identify, accept ownership, and manage assimilation of other software products.

Existing software may be assimilated into a project such that the project team accepts responsibility for maintaining, supporting, and potentially continuing development of the software. Assimilation shall consider the effort needed to ensure that the software meets the project’s verification and other software quality practices and standards. Assimilation shall also consider the potential impact to the project’s mission, applicable practices, and LOF.

Artifacts

AR11 Design artifacts [documentation and/or reviews] (PR13)

AR12 Implementation artifacts [software code, assimilated other software, design documents, user documentation, developer’s guide, installation guide, theory manual, interface manual, etc.] (PR14, PR16)

AR13 Identification and acquisition records (PR15, PR16)

Example Inputs

- External knowledge (subject matter experts, algorithms, technical reports) (PR13, PR14)
- Assimilated software (from a source outside the project) (PR14)
- Third party software (PR15)
- Other software (PR16)

AR3 Approved project processes (PR13, PR14)

AR6 Product expectations and requirements (PR13, PR14)

AR7 Software requirements and attributes (PR13, PR14)

AR8 List of stakeholders and organizational commitments (PR13)

AR9 Project plan (PR13, PR14)

AR13 Identification and acquisition records (PR13, PR14)

AR16 Managed issues [enhancements, defects, inquiries] (PR13, PR14)

AR17 Release specification (PR14)

AR22 Software verification plan (PR15, PR16)

4.2.2 Configuration Management Process Area

Overview Description

The purpose of configuration management (CM) is to provide a controlled environment for development, production, and support activities. CM includes identifying which software product artifacts are to be managed; maintaining version controlled baselines of these artifacts; providing an issue tracking system for recording associated issues or change requests related to product artifacts; and tracking the status of these issues throughout the project’s lifetime. CM ensures retrieval of any baselined artifact over the project’s lifetime. Some specific artifacts (for example,

records) and their retention schedule shall be subject to ASC organizational procedures as well as SNL's Record Management Policies, <http://www-irn.sandia.gov/recordsmgmt/rmm/rmmframe.html>. Projects shall follow organizational procedures for the identification, collection, organization, filing, storage, maintenance, retrieval, distribution, retention, and disposition of such records.

Practices

PR17. Perform version control of identified software product artifacts.

As part of version control project teams identify project artifacts that shall be maintained in a repository, implement access and version control those artifacts, create and recover product baselines, and manage changes to these baselines.

PR18. Record and track issues associated with the software product.

Issue tracking (i.e., problem reporting and corrective action) is a cross-cutting dimension of this Software Quality Plan and shall be addressed across all ASC SQE practices. All potential team issues, whether originated internally or externally, may affect the product and shall be captured using this practice. This practice shall include a process (i.e., change management) of recording and tracking all appropriate changes that occur to identified software product artifacts, including requirements, throughout their lifetime. Issue tracking shall include an issue classification scheme and allow for the submittal of enhancement requests, problem and defect reports, inquiries, and other project related issues. Customers are a source of submitted issues. Section 4.2.4, Deployment and Lifecycle Support, addresses customer issue submission and response.

PR19. Ensure backup and disaster recovery of software product artifacts.

This practice ensures backup is performed and disaster recovery of software product artifacts and associated baselines is possible if the repository becomes unavailable or is destroyed. Backup and recovery capability shall include the identification of where product artifacts are stored, a defined schedule for when backups are made, and a method of recovering or restoring backups if a disaster occurs. The disaster recovery capability shall be tested annually to ensure that artifacts can be recovered and restored with minimal disruption to other project activities. This practice, if performed by a source outside the project team, shall be satisfied through confirmation that system administration is performing backups, ensuring safe storage, and testing recovery. See supplemental practice guidance regarding backup and disaster recovery in SPG19.

Supplemental Practice Guidance

SPG19 Backup and disaster recovery guidance: Advanced Simulation and Computing (ASC) Program Guide to Backup and Disaster Recovery²⁰ available at:
https://wfsprod01.sandia.gov/intradoc-cgi/idc.cgi_isapi.dll?IdcService=GET_SEARCH_RESULTS&QueryText=dDocName=WFS557589

Artifacts

AR14 Version controlled records, including baselines and associated configurations (PR17)

AR15 Backup records and recovery test results (PR19)

AR16 Managed issues [process issues, product quality results (for example, non-conformances), enhancements, defects, inquiries] (PR18)

Example Inputs

- Customer issues (PR18)

AR1-AR27 Appropriate product artifacts (PR17, PR18)

4.2.3 Product Integration Process Area

Overview Description

The purpose of product integration is to manage the integration of product components and interfaces into verified versions of the software product that is distributed to customers. Product integration includes determining a sequence for integrating the software product and its components, determining an environment for carrying out the integration, and developing procedures and criteria for being able to certify that the integration is complete and successful. Product integration involves release management, defined as handling the requests for a release as well as preparation of the release. A release shall include all elements of the product or a defined subset of the product. When the project team has completed all artifacts necessary for a release, the team creates a baseline in preparation for distribution. The baselined product undergoes release certification before being distributed and supported. Release certification ensures that all release criteria are satisfied, identified release artifacts are adequately reviewed, and all planned testing is completed and satisfactory. Product integration planning is also addressed in other stages of the product lifecycle. It begins with the practices in 4.1.4, Requirements Development; and Management, is further addressed and planned for with the practices in 4.1.5, Risk Management; and 4.1.6, Project Planning and Oversight, and is verified with the practices in 4.3, Software Verification.

Practices

PR20. Plan and generate the release package.

Prior to planning for what a given release will include, the project team shall consider the sequence of activities needed to create a release; provide a suitable environment (for example, test equipment, simulators, hardware, tools, and devices for recording logs of results) for performing the integration; and establish integration procedures (for example, a release checklist). This practice shall include determination of the release criteria; such as the release contents, dependencies on external products, targeted distribution date, required resources, and internal activities for completion of the release. Examples of release contents include code, user guides, training material, theory manuals, installation notes, and test cases that the customer can run to check installation. Internal activities shall include reviews of all release contents before any major product release, installation testing, and generation of release notes. Release notes shall include history of other releases associated with the project.

PR21. Certify that the software product (i.e., code and its related artifacts) is ready for release and distribution.

Certification of the software product requires reviewing and confirming that all product components and interfaces are available, complete, compatible, and function according to their specifications. Release certification ensures that all release criteria identified in planning for the release are satisfied. Certification may be a multi-step process verifying the release sufficiently to justify distribution. A final review before each release shall verify that all required artifacts exist and are associated with the correct version number.

Artifacts

AR17 Release specification (PR20)

AR18 Product release package (bill of materials, release notes, certification, software, etc.) (PR20, PR21)

Example Inputs

- Internal/external request for a release (PR20)
- Information for release notes (PR20)
- Product artifacts that will be included in the release (PR20)

AR23 Test artifacts [test cases, test results, test reports] (PR21)

AR24 Technical reviews (PR21)

4.2.4 Deployment and Lifecycle Support Process Area

Overview Description

The purpose of this process area is to provide verified product releases to customers and then to assist and train those customers in the installation, operation, and ongoing use of the product. If data or databases are a delivered project component, then procedures for testing and documenting them shall be developed and included in the delivered product. Deployment and lifecycle support also includes those activities required to manage feedback concerning the product. Each project team shall define and implement a customer support plan to address the needs and expectations of appropriate customers (for example, those customers with organizational commitments). The customer support plan may be a single shared agreement intended to address the needs of multiple customers. Resources for implementing this plan shall factor into project planning.

Practices

PR22. Distribute release to customers.

The project team shall determine an effective means for packaging and delivering the assembled product release, which includes satisfying applicable requirements and standards for the customers' production environment. In distributing the release to customers, the project team shall consider whether any license agreements need to be updated, whether the product falls under export control restriction, and whether certain types of customers (e.g., end users, funding sources, other software projects, etc.) need special instructions or support. The project team shall also notify appropriate customers that a previous version of the product is being retired.

PR23. Define and implement a customer support plan.

A customer support plan shall specify the period of support, responsibilities, point of contact for questions on any aspect of the product release, commitment to deliver documentation and training, and other support deemed necessary. The support plan shall include a feedback process for the submittal, prioritization, and timely resolution of issues associated with the product. The feedback process shall use the issue tracking process defined in configuration management. The customer support plan shall also include information related to product installation, supported platforms, consistent product interfaces, and frequency of product installations. Customers who intend to provide funding for support activities shall be included in negotiations regarding what will be included in the plan.

PR24. Implement the training identified in the customer support plan.

During requirements gathering the project team shall determine the requirements of the product training plan that includes requisite documentation. Example topics covered by training include installation, use, theory manuals, tutorials, and tests. Project teams shall maintain records (such as class, attendees, dates, and training evaluations) for training they deliver.

PR25. Evaluate customer feedback to determine customer satisfaction.

The ultimate measure of quality is customer satisfaction. At the appropriate point in the product's lifecycle, the project team shall solicit customer feedback regarding the level of satisfaction with the product and the support the team provides. This information supports identification of systemic quality problems and opportunities for process improvement.

Artifacts

- AR16** Managed issues [enhancements, defects, inquiries] (PR23, PR24, PR25)
- AR18** Product release package (bill of materials, release notes, certification, software, etc.) (PR22)
- AR19** Customer support plan including training (PR23)
- AR20** Customer training records (PR24)
- AR21** Customer satisfaction evaluation (PR25)

Example Inputs

- Identified customers for whom release is intended (PR22)
 - List of target platforms for the release (PR22)
 - Information for release notes (PR22)
 - Product artifacts that will be included in the release (PR22)
- AR6** Product expectations and requirements (PR23, PR24)
- AR8** List of stakeholders and organizational commitments (PR23, PR24, PR25)

4.3 Software Verification SQE Category

Some ASC project teams have participated in the development of a verification and validation (V&V) plan and, perhaps, have performed some of the activities outlined in this plan. Information from an existing V&V plan can potentially be leveraged for the software verification practices. V&V plans shall include the test planning related to a verification test suite and technical reviews. If a project team has a test plan but no software verification plan, the test plan can be enhanced with planning information for technical reviews.

4.3.1 Software Verification Process Area

Overview Description

The purpose of software verification is to ensure that requirements are accurately, correctly, and completely implemented. Software verification also ensures product characteristics necessary for safe and proper use of the software are addressed. Software verification occurs throughout the entire product lifecycle.

Software verification activities are an integral part of software development, deployment, operation, and lifecycle support practices. In this context, the goal is to detect actual or potential problems as early as possible. Typically, software artifacts subject to verification include specifications, requirements, design, code, third-party libraries, software verification plan, test cases and results, product documentation, and training package(s). If these artifacts are changed, retesting and reevaluation of the changes are required.

Software verification determines that the project team has correctly implemented the requirements. In some phases of the product lifecycle, software verification relies on technical reviews of various work products; however, once code is developed and integrated, testing becomes the focus with associated reviews of test results.

Verification and validation are best addressed in an application-specific context. In this respect, validation of the code by comparison to physical data is the responsibility of the analysis team

and will not be addressed as part of these code development SQE practices. However, there is an element of software testing that is aligned to code development activities. The first two of the following practices address software testing.

Practices

PR26. Develop and maintain a software verification plan.

Projects shall identify the list of artifacts to be reviewed, a list of knowledgeable reviewers, test and technical review approach, tools, associated verification test cases, required schedules for tests and technical reviews, required resources, and required responsibilities. The software verification plan shall define tests and reviews of code that demonstrate that requirements are being met and acceptance criteria that are used in the review of test results and code. The software verification plan shall specify the types of tests, when test results are reviewed, the technical reviews to be performed and their objectives, and the technical review schedule. See the ASC SQE website¹⁶ for additional information.

PR27. Conduct tests to demonstrate that acceptance criteria are met and to ensure that previously tested capabilities continue to perform as expected.

Testing occurs throughout the product lifecycle. Tests and results from performing tests found in the software verification plan or in separate test cases shall be reviewed with respect to each test's associated acceptance criteria. Test cases shall be reviewed when they are initially created and after any changes to code related to the tests. See the ASC SQE website¹⁶ for a discussion of test terms and test categories.

PR28. Conduct independent technical reviews to evaluate adequacy with respect to requirements.

These reviews apply to all software artifacts requiring review per this Software Quality Plan and shall include evaluation of adequacy with respect to intended use and acceptance criteria. Examples of acceptance criteria include comparison of tests with analytic solutions or other pedigreed codes, traceability analysis to determine support of the requirements for each critical artifact, interface analysis to check consistency and completeness of the user interface, data flow such as unit conversion, and control flow between components represented by the artifact.

Independent technical reviews shall include some participants that are independent of the creation of the item or activity being reviewed and knowledgeable in relevant subject areas.

Artifacts

AR16 Managed issues [product quality results (for example, non-conformance), enhancements, defects, inquiries] (PR27, PR28)

AR22 Software verification plan (PR26)

AR23 Test artifacts [test cases, test results, test reports] (PR27)

AR24 Technical reviews (evidence that review occurred and review results) (PR28)

Example Inputs

AR1 Strategic plan [project's mission, management, stakeholders] (PR26)

AR6 Product expectations and requirements (PR26, PR27, PR28)

AR7 Software requirements and attributes (PR26, PR27, PR28)

AR8 List of stakeholders and organizational commitments (PR26, PR27, PR28)

AR11 Design artifacts (PR27, PR28)

AR12 Implementation artifacts [including assimilated other software] (PR27, PR28)

4.4 Training Support Category

Training spans the three primary SQE categories of Project Management, Software Engineering, and Software Verification (see Figure 2) and addresses the importance of the “human asset” in ASC software development.

4.4.1 Training Process Area

Overview Description

The goal of this training process area is to enhance the skills of a staff that is already highly trained and educated in the areas of scientific software development, algorithms, and/or computer science. Training develops the skills and knowledge of individuals and teams so they can execute effectively and efficiently their project technical and process roles and responsibilities. Project teams shall ensure that the training needs of the project are satisfied in accordance with their project plan. Customer training is addressed in Section 4.2.4 Deployment and Lifecycle Support.

Required Corporate training (e.g., annual security training, ES&H awareness, export control awareness, etc.) is the responsibility of SNL, not the individual projects, and is therefore excluded from this process area.

Implementing these two training practices shall include preparing a list of required and desired training to be taken, identifying when the training is needed and the acceptable methods of receiving the training (for example, mentoring, classroom setting, online course, etc.), documenting when the training is actually taken, and developing methods for gauging the effectiveness of the training.

Training is a cross-cutting dimension of this Software Quality Plan and shall be addressed across all ASC SQE practices. A well-implemented and purposeful training plan contributes to institutionalizing the Software Quality Plan across the entire project team. When applicable, the ASC Program shall plan and coordinate training efforts across the Program to provide cost-effective means of addressing needed training across multiple projects.

Practices

PR29. Determine project team training needed to fulfill defined roles and responsibilities.

Training needs shall be determined by comparing the actual skills and knowledge of the team members to the skills and knowledge necessary to complete their project roles and responsibilities.

PR30. Track training undertaken by project team.

The project team shall maintain training records indicating which training the project team members participated in, when the training occurred, and the measurements and/or impact assessments associated with the training.

Artifacts

AR25 Project team training needs (PR29)

AR26 Project team training records (PR30)

Example Inputs

- Organization training requirements and opportunities (PR29)
- AR3** Approved project processes (PR29)

AR9 Project plan [task list, resource information, roles and responsibility assignments] (PR29)

4.5 Practices and Generated Artifacts Summary Table

Table 3 provides a joint list of the 30 SQE practices and the required artifacts generated by each practice.

Table 3. Practices and Generated Artifacts

PR #	Practice Description	
	SPG#	Supplemental Practice Guidance (Note: not all practices have supplemental practice guidance.)
	AR #	Artifact Description
PR1	Document and maintain a strategic plan.	
	AR1	Strategic plan [project's mission, management, stakeholders, stakeholder roles and responsibilities, team operating procedures]
PR2	Perform a risk based assessment, determine level of formality and applicable practices, and obtain approvals.	
	SPG2a	Risk-based assessment procedure to determine LOF (template): ASC Risk-Based Assessment Procedure ¹⁸
	SPG2b	ASC safety software guidance: Instructions for Determining Whether an ASC Software Product Should be Categorized as DOE Order 414.1c Safety Software ¹⁹
	AR2	Approved level of formality and applicable practices [tailoring and/or waivers]
PR3	Document, monitor, and control lifecycle processes and their interdependences, and obtain approvals.	
	AR3	Approved project processes
	AR16	Managed issues: [process issues, product quality results (for example, non-conformances), enhancements, defects, inquiries]
PR4	Define, collect, and monitor appropriate process metrics.	
	AR4	Process and product metrics [defined metrics, quantitative objectives for process performance, statistically managed measurements for a subprocess]
	AR16	Managed issues: [process issues, product quality results (for example, non-conformances), enhancements, defects, inquiries]
PR5	Periodically evaluate quality issues and implement process improvements.	
	AR5	Project process improvement actions
	AR16	Managed issues [process issues, product quality results (for example, non-conformances), enhancements, defects, inquiries]
	AR27	Self-assessment and independent appraisal results
PR6	Identify stakeholders and other requirements sources.	
	AR6	Product expectations, requirements, and constraints
	AR8	List of stakeholders and organizational commitments
PR7	Gather and manage stakeholders' expectations, requirements, and constraints.	
	AR7	Software requirements and attributes
PR8	Derive, negotiate, manage, and trace requirements.	
	AR6	Product expectations, requirements, and constraints
	AR7	Software requirements and attributes
	AR8	List of stakeholders and organizational commitments
PR9	Identify and analyze risk events.	
	AR9	Project plan [risk events, risk plan]
PR10	Define, monitor, and implement the risk response.	
	AR10	Project reviews and needed (corrective) actions [risk responses]
PR11	Create and manage the project plan.	
	AR9	Project plan [risk events, risk plan, overview, milestones, task list, resource information, roles and responsibility assignments, assumptions, constraints, dependencies, budget, schedule, SCM plan, etc.]

Table 3. Practices and Generated Artifacts (continued)

PR #	Practice Description	
	SPG#	Supplemental Practice Guidance (Note: not all practices have supplemental practice guidance.)
	AR #	Artifact Description
PR12	Track project performance versus project plan and implement needed (i.e., corrective) actions.	
	AR10	Project reviews and needed (corrective) actions [risk responses, tracking and oversight responses]
PR13	Communicate and review design.	
	AR11	Design artifacts [documentation and/or reviews]
PR14	Create required software and product documentation.	
	AR12	Implementation artifacts [software code, assimilated other software, design documents, user documentation, developer's guide, installation guide, theory manual, interface manual, etc.]
PR15	Identify and track third party software products and follow applicable agreements.	
	AR13	Identification and acquisition records
PR16	Identify, accept ownership, and manage assimilation of other software products.	
	AR12	Implementation artifacts [assimilated other software.]
	AR13	Identification and acquisition records
PR17	Perform version control of identified software product artifacts.	
	AR14	Version controlled records, including baselines and associated configurations
PR18	Record and track issues associated with the software product.	
	AR16	Managed issues [process issues, product quality results (for example, non-conformance), enhancements, defects, inquiries]
PR19	Ensure backup and disaster recovery of software product artifacts.	
	SPG19	Backup and disaster recovery guidance: Advanced Simulation and Computing (ASC) Program Guide to Backup and Disaster Recovery ²⁰
	AR15	Backup records and recovery test results
PR20	Plan and generate the release package.	
	AR17	Release specification
	AR18	Product release package (bill of materials, release notes, certification, software, etc.)
PR21	Certify that the software product (i.e., code and its related artifacts) is ready for release and distribution.	
	AR18	Product release package (bill of materials, release notes, certification, software, etc.)
PR22	Distribute release to customers.	
	AR18	Product release package (bill of materials, release notes, certification, software, etc.)
PR23	Define and implement a customer support plan.	
	AR16	Managed issues [process issues, product quality results (for example, non-conformance), enhancements, defects, inquiries]
	AR19	Customer support plan including training
PR24	Implement the training identified in the customer support plan.	
	AR16	Managed issues [process issues, product quality results (for example, non-conformance), enhancements, defects, inquiries]
	AR20	Customer training records
PR25	Evaluate customer feedback to determine customer satisfaction.	
	AR16	Managed issues [process issues, product quality results (for example, non-conformance), enhancements, defects, inquiries]
	AR21	Customer satisfaction evaluation
PR26	Develop and maintain a software verification plan.	
	AR22	Software verification plan
PR27	Conduct tests to demonstrate that acceptance criteria are met and to ensure that previously tested capabilities continue to perform as expected.	
	AR16	Managed issues [process issues, product quality results (for example, non-conformance), enhancements, defects, inquiries]
	AR23	Test artifacts [test cases, test results, test reports]
PR28	Conduct independent technical reviews to evaluate adequacy with respect to requirements.	
	AR16	Managed issues [process issues, product quality results (for example, non-conformance), enhancements, defects, inquiries]
	AR24	Technical reviews (evidence that review occurred and review results)
PR29	Determine project team training needs to fulfill assigned roles and responsibilities.	
	AR25	Project team training needs
PR30	Track training undertaken by project team.	

PR #	Practice Description	
	SPG#	Supplemental Practice Guidance (Note: not all practices have supplemental practice guidance.)
	AR #	Artifact Description
	AR26	Project team training records

5.0 STRATEGY FOR CONFORMANCE TO SQE PRACTICES

The ASC Program uses self-assessments, independent appraisals, and external audits to evaluate the degree to which the ASC projects have implemented the SQE practices, identify best practices applied by the ASC projects, and identify process improvement activities. Within the context of this document, a self-assessment is defined as an appraisal that an organization performs internally for the purposes of process improvement. An independent appraisal is defined as an examination of one or more processes by a trained team of professionals who are external to the appraised organization or project, but not necessarily external to SNL. This examination uses an appraisal reference model (a defined standard to which the organization or project shall be appraised) as the basis for determining, at a minimum, strengths and weaknesses. An external audit is defined as an examination of one or more processes by a team of professionals who are external to SNL. The following elements shall be used in appraisals of a project team's process implementation and achievement of the SQE practices identified in this Software Quality Plan:

- ASC Program management sponsors appraisals and defines the overall goals and objectives for each appraisal.
- For independent appraisals, the appraisal sponsor creates an independent team to develop an appropriate approach and appraisal instrument to achieve the stated appraisal goals and objectives. The independent appraisal team shall be technically qualified and knowledgeable according to education, training, and experience.
- The appraisal sponsor authorizes an independent team to perform appraisals.
- Prior to the independent appraisal, a project team may perform a self-assessment if one previously has not been completed.
- An external independent appraisal is conducted when such an assessment provides a business or mission advantage to the organization or its customers. For further guidance, refer to the SNL CPR001.3.10: Corporate Self-Assessment Process²¹.
- Results of self-assessments and independent appraisals are documented and presented to AQMC and to the appraisal sponsor.
- The appraisal sponsor communicates best practices identified from the appraisals to AQMC and to the appraised project teams.

Appraisal artifacts include the documented appraisal procedure as well as the documented appraisal report. In general, documented appraisal reports shall

- describe the appraisal objectives, scope, approach, and performance requirements and quantitative criteria;
- identify appraisers and persons contacted;
- identify documents, material, operations, activities, and conditions appraised;
- present issues observed; and
- summarize the extent of achievement and performance relative to appraisal scope, performance requirements, and associated criteria.

SNL ASC management shall direct project teams to perform internal self-assessments to compare their current practice implementations to management-defined goals and associated criteria. This approach will help the software teams to determine those areas in which they are making good progress and where they need to focus improvement efforts.

Project teams involved in independent appraisals shall focus on ensuring that documented processes for the various practices are accessible and followed. The teams shall furnish project artifacts that demonstrate that the project's defined processes are being followed. Team members involved in appraisal interviews will be asked to explain how the project operates, whether processes are in place, and how consistently they are following these processes. The "ASC Appraisal Evidence Collection Instrument" is included on the ASC SQE website²². This instrument is applicable for both self-assessments and independent appraisals.

REFERENCES

Text References: The following references are numbered according to the order in which they appear in the body of this document.

1. Boucheron, E, R. Drake, H.C. Edwards, M. Ellis, C. Forsythe, R. Heaphy, A. Hodges, C. Pavlakos, J. Schofield, J. Sturtevant, and C.M. Williamson, Sandia National Laboratories Advanced Simulation and Computing (ASC) Software Quality Plan: Part 1: ASC Software Quality Engineering Practices, Version 1.0, SAND2004-6602, Sandia National Laboratories, January 2005. Available at: http://infoserve.sandia.gov/sand_doc/2004/046602.pdf
2. Boucheron, E, R. Drake, H.C. Edwards, C. Forsythe, P. Hackney, R. Heaphy, A. Hodges, M. Minana, C. Pavlakos, J. Schofield, J. Sturtevant, and C.M. Williamson, Sandia National Laboratories Advanced Simulation and Computing (ASC) Software Quality Plan: Part 1: ASC Software Quality Engineering Practices, Version 2.0, SAND2006-5998, Sandia National Laboratories, September 2006. Available at: <https://wfsprod01.sandia.gov/groups/srn-uscitizens/documents/document/wfs400032.pdf>
3. Zepper, J., K. Aragon, M. Ellis, D. Eaton, and K. Byle, ASCI Applications Software Quality Engineering Practices, Version 2, SAND2003-0962, Sandia National Laboratories, April 2003. Available at: http://infoserve.sandia.gov/sand_doc/2003/030962.pdf
4. Berg, R. and A. Hodges, ASCI Simulation and Computer Science (S&CS) and Ongoing Computing Software Quality Plan, Version 2, March 2003. Available at: <https://wfsprod01.sandia.gov/groups/srn-uscitizens/documents/document/wfs125757.pdf>
5. Ellis, M., C. M. Williamson, J. Schofield, and L. Bonano, 2003 SNL ASCI Applications Software Engineering Assessment Report, SAND2004-0075, Sandia National Laboratories, February 2004. Available at: http://infoserve.sandia.gov/sand_doc/2004/040075.pdf
6. Pilch, M., T. Trucano, J. Moya, G. Froehlich, A. Hodges, and D. Peercy. Guidelines for Sandia ASCI Verification and Validation Plans - Content and Format: Version 2.0, SAND2000-3101, Sandia National Laboratories, January 2001. Available at: http://infoserve.sandia.gov/sand_doc/2000/003101.pdf
7. Hodges, A. L., G. K. Froehlich, M. Pilch, and D. E. Peercy, Risk Management Plan, Sandia National Laboratories ASCI V&V Program, SAND2002-1048, Sandia National Laboratories, April 2002. Available at: http://infoserve.sandia.gov/sand_doc/2002/021048.pdf
8. CMMI for Development, v1.2, Software Engineering Institute, August 2006. CMMI Website available at: <http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf>
9. IEEE Std.610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, Standards Coordinating Committee of the IEEE Computer Society, USA, December 2000. Available via: <http://ieeexplore.ieee.org/servlet/opac?punumber=2238>
10. ISO 9000:2000 Quality Management Systems – Fundamentals and Vocabulary and ISO 9001:2000 Quality Management Systems – Requirements, International Standards Organization, December 2000. Available through SNL by clicking on “Standards and Guidelines” (on left menu) at: <http://www-irn.sandia.gov/ilms/data/ISO/>
11. Corporate Process Requirement No. CPR001.3.6, Corporate Software Engineering Excellence, Sandia National Laboratories, October 2007. Available at: http://www-irn.sandia.gov/policy/leadership/software_quality.html

12. Corporate Process Requirement No. CPR001.3.2, Corporate Quality Assurance Program, Sandia National Laboratories, August 2003. Available at: <http://www-irn.sandia.gov/policy/leadership/quality.html>
13. Department of Energy/National Nuclear Security Agency, DOE/NNSA Quality Criteria (QC-1), Revision 10, February 10, 2004. Available at: <http://prp.sandia.gov/documents/qc-1.asp>
14. Department of Energy Order 414.1C, Quality Assurance, June 17, 2005. Available at: <http://www.directives.doe.gov/pdfs/doe/doetext/neword/414/o4141c.html>
15. Hodges, A., G. Froelich, D. Peercy, M. Pilch, J. Meza, M. Peterson, J. LaGrange, L. Cox, K. Koch, N. Storch, C. Nitta, and E. Dube, Department of Energy, *ASCI Program Software Quality Engineering: Goals, Principles, and Guidelines*, DOE/DP/ASC-SQE-2000(P)FDRFT-VERS2, Albuquerque, NM, February 2001. Available at: http://infoserve.sandia.gov/sand_doc/2001/010253p.pdf
16. ASC Software Quality Engineering website, available at <http://www-irn.sandia.gov/ascsqe>
17. Turgeon, Jennifer, M. Minana, Sandia National Laboratories Advanced Simulation and Computing (ASC) Appraisal Method for Implementation of the ASC Software Quality Engineering Practices, Version 1.0, SAND 2008-0403, Sandia National Laboratories, February 2008. Available at: <https://wfsprod01.sandia.gov/groups/srn-uscitizens/documents/document/wfs454296.pdf>
18. Minana, M., J. Turgeon, and M. Pilch, ASC Risk-Based Assessment Procedure, Sandia National Laboratories, May 2008. Available at: https://wfsprod01.sandia.gov/intradoc-cgi/idc_cgi_isapi.dll?IdcService=GET_SEARCH_RESULTS&QueryText=dDocName=WFS680941
19. M. Minana, M., D. Peercy, and P. Hackney, Instructions For Determining Whether An ASC Software Product Should Be Categorized As DOE Order 414.1c Safety Software, Sandia National Laboratories, April 2008. Available at: https://wfsprod01.sandia.gov/intradoc-cgi/idc_cgi_isapi.dll?IdcService=GET_SEARCH_RESULTS&QueryText=dDocName=WFS710561.
20. Baldonado, E., M. Minana, and J. Turgeon, Advanced Simulation and Computing (ASC) Program Guide to Backup and Disaster Recovery, Sandia National Laboratories, January 2008. Available at: https://wfsprod01.sandia.gov/intradoc-cgi/idc_cgi_isapi.dll?IdcService=GET_SEARCH_RESULTS&QueryText=dDocName=WFS557589
21. Corporate Process Requirement No. CPR001.3.10, Corporate Self-Assessment Process, Sandia National Laboratories, August 2003. Available at: <http://www-irn.sandia.gov/policy/leadership/cpr001.3.10.html>
22. ASC Software Quality Engineering website, "Assessments," "FY07-08 Software Quality Appraisals," available at <http://www-irn.sandia.gov/ascsqe/fy07-08appraisals.html>
23. Minana, M., J. Turgeon, P Hackney, and M. Pilch, "Mapping ASC Software Quality Plan to Customer Directives," available at <https://wfsprod01.sandia.gov/groups/srn-uscitizens/documents/document/wfs672480.pdf>.
24. Hodges, A., D. Peercy, and J. Schofield, Specific Use Specification, Sandia Software Quality Assurance Program (SS-R89727), Sandia National Laboratories, September 2007. Available at: <https://wfsprod01.sandia.gov/intradoc->

https://wfsprod01.sandia.gov/intradoc/cgi/idc.cgi?isapi.dll?IdcService=GET_SEARCH_RESULTS&QueryText=dDocName=WFS528492

Additional References: The following documents are not referenced in the body of this document but provide additional information that is useful in developing and implementing Sandia ASC SQE policies and practices:

1. Percy, D., A. Hodges, M. Minana, and J. Schofield, Specific Use Specification, Software Quality Implementation Group (SQUIG) Self-Assessment Instrument for CPR001.3.6 (SS-R94286), Sandia National Laboratories, September 2007. Available at:
https://wfsprod01.sandia.gov/intradoc/cgi/idc.cgi?isapi.dll?IdcService=GET_SEARCH_RESULTS&QueryText=dDocName=WFS528493
2. Program Management Institute (PMI) Standard: A Guide to the Project Management Book of Knowledge (PMBOK Guide) 2000 Edition, Project Management Institute, Newtown Square, Pennsylvania.

This page intentionally left blank.

GLOSSARY AND ACRONYMS

Glossary

acceptance criteria The criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity.

artifact A documented process, deliverable or work product. A configuration-controlled artifact is stored in a corporate repository (library) and changes to it are controlled via reported issues.

assessment An appraisal by a trained team of software professionals to determine the state of an organization's current software process, to determine the high-priority software process-related issues facing that organization, and to obtain the organizational support for software process improvement.

baseline A set of specifications or artifacts that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through change control procedures.

benchmarking A quality tool used to periodically and continually measure and compare an organization's work processes with those in competing or similar organizations. The goal of benchmarking is to increase the organization's performance by adopting the best practices of industry leaders.

best practices Those activities that have proven to be of high value, have improved quality, have improved productivity, or have enhanced customer satisfaction. Typically, these practices are measured activities or have metrics to show their value and are leveraged across an organization.

critical path method A network analysis technique used to predict project duration by analyzing which sequence of activities (which path) has the least amount of scheduling flexibility.

customer The party (individual, project, or organization) responsible for accepting the product or for authorizing payment. The customer may be an end-user of the product. Customers are a subset of stakeholders.

customer support The assistance, training, and documentation a project provides to ensure that the customer is satisfied and able to use the product as intended. Typically, a support plan is drawn up to specify what will, and what will not, be provided by the project team and for what period of time.

defect 1) A flaw in a system or system component that causes the system or component to fail to perform its required function. 2) Non-fulfillment of a requirement related to an intended or specified use.

design of experiments An investigation carried out in a planned manner and which relies on a statistical assessment of results to reach conclusions at a stated level of confidence.

error-proofing Also known as fool-proofing, mistake-proofing, and Poka-Yoke (Japanese quality term) An example of error-proofing for software development is a process checklist. The checklist prevents errors from missing an activity or performing the activity in the wrong sequence.

Gantt A graphic display of schedule-related information (sometimes called a bar chart).

interface analysis The evaluation of presentation and flow (control and data) between components represented by the artifact.

issue A point of concern, a problem, or a comment that is raised in regard to a practice of a software lifecycle area. The issue is a form of feedback and will usually be specific to an artifact suggesting rework, improvement, or enhancement.

level of formality The degree of detail, form, and frequency to which a project defines and carries out its process for implementing a practice.

lifecycle The period of time that begins when a software product is conceived and ends when the software is no longer available for use. Typically a lifecycle includes concept, requirements, design, implementation, test, installation, and operation and maintenance phases. These phases may overlap or be performed iteratively.

lifecycle model An approach to the lifecycle that provides adequate detail of the order and phases. Some examples include spiral, evolutionary, sequential, and iterative.

measure A unit of measurement (such as source lines of code or document pages of design).

measurement The dimension, capacity, quantity, or amount of something (for example, 300 source lines of code or seven document pages of design).

metric A quantitative measure of the degree to which a system, component, or process possesses a given attribute.

mitigate Reduce the probability and/or impact of a risk to below an acceptable threshold.

policy An accepted principle, established by decision makers, to direct and influence the activities of those to whom the policy pertains.

practice A set of activities identified for accomplishing some portion of the required areas identified in the Software Quality Plan.

process A set of steps performed for a given purpose (for example, implementation of a practice). A well-documented process contains inputs, outputs, roles and responsibilities, sequences and dependencies, reviews and approvals, and entry and exit criteria. A process will have many but not necessarily all these attributes. It may be textual or graphical but must not be merely imaginary or virtual.

process metric This type of metric measures the characteristics of the overall development process, such as the number of defects found throughout the process during different kinds of reviews.

product metric This type of metric is a measurement of an intermediate or final product of software development and, therefore, addresses the output of a software development activity. Examples of such metrics are a size metric for the number of requirements and a complexity metric for software.

production software This type of software is implemented in a production environment, characterized as stable (meaning changes are recorded and analyzed), and fully supported by the project development team.

program evaluation and review technique An event-oriented network analysis technique used to estimate program duration when there is uncertainty in the individual activity duration estimates.

quality 1) Conformance to customer requirements and expectations. 2) The degree to which a system, component, or process meets specified requirements. 3) The degree to which a system, component, or process meets customer or user needs or expectations.

regression test Selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements.

release A snapshot in time of a software product available for distribution. Typically includes software as source or executable.

release plan A plan prepared and followed by the project team specifying what needs to be accomplished for releasing the next version of a software product. The release plan typically specifies what the release will contain; what the release depends on externally such as compilers, version of required utility, etc.; when the release will be ready for distribution; what resources will be needed to prepare the release; and other dependencies for completing the release (for example, installation testing, user documentation, reviews, training, and release notes).

requirement A need or expectation that is stated, generally implied, or obligatory.

review A quality assurance activity that establishes confidence in codes and supports software verification. Types of reviews are as follows:

- **management** An evaluation performed to verify that commitments for the specified activities have been satisfied.
- **quality** An evaluation performed to verify achievement of process and artifact requirements.
- **technical** An evaluation to determine if the content of the item submitted for review conforms to the requirements.

reviewer An independent person qualified to perform a review.

risk A combination of the likelihood of an event's occurrence and its impact.

risk mitigation Reduce the probability and/or impact of a risk to below an acceptable threshold and/or increasing the positive consequence.

risk plan This document details all identified risks including description, cause, probability of occurring, impact(s) on objectives, proposed responses, owners, and current status. The plan also addresses procedures and techniques to enhance opportunities and to reduce threats to the projects' objectives.

role A set of defined responsibilities that are assumed by one or more individuals.

software engineering The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software.

software product Any software project deliverable given to the customer. One software product is typically the code (executable and/or source).

software quality assurance 1) A set of activities designed to evaluate the process by which products are developed. 2) Planned and systematic actions necessary to provide adequate confidence that the software product conforms to established requirements.

software quality engineering The practices a software team follows to ensure that quality standards are incorporated into its software product.

software verification 1) Ensures that requirements are accurately, correctly, and completely (with respect to the delivery commitments) implemented throughout the entire product lifecycle, and that requirements are adequate from the intended uses of the software. 2) The process of determining whether or not the mathematical formulation is solved correctly, that is, whether the computer simulation correctly represents the conceptual model and its solution.

stakeholder A group or individual that is affected by or is in some way accountable for the outcome of an undertaking. Stakeholders may include project members, suppliers, customers, end users, and others.

subject matter expert An individual who is responsible for providing guidance and information to the software project team in areas or topics outside the scope of the team's expertise.

supplier An organization that supplies materials, goods or services directly or indirectly for a customer.

system requirements The conditions or capabilities that must be met or possessed by a system or system component to satisfy a condition or capability needed by a user to solve a problem.

test case Each test must have a specification that contains information to identify the test, test environment, test procedure, and expected test results with acceptance criteria. An automated test will typically capture this information in the script.

test plan A description of the technical and management approach to be followed for testing a system or component. Typical contents identify the items to be tested, features to be tested, any risks requiring mitigation, tasks to be performed, responsibilities, schedules, required resources for the testing activity, and reference to test cases. The plan must identify the types of tests that will be conducted as well as any additional tests that are needed to provide confidence that the software product does not contain any defects and to demonstrate that requirements are met.

test results Output generated as a consequence of executing test cases. Examples of test results include logs generated manually or by automated scripts, issues identified during test and evaluation activities, test and evaluation summary report describing if/how activities deviated from the plan, summarizing results, and providing recommendations. An important element of test results is that each test case maps to its corresponding test output and that the date and time are recorded.

third party product A third party product is an application or library used or required by a SNL ASC code application; however, ASC project teams do not normally maintain this particular software. Many of these third party product sets are developed at Sandia while other sets are developed by other government labs, commercial vendors, and university partners.

traceability 1) The degree to which a relationship can be established between two or more artifacts of the product lifecycle, especially artifacts having a predecessor (successor or master) subordinate relationship to each other. 2) Ability to trace history, application, or location of that which is under consideration.

traceability analysis Evaluation to determine support of the requirements for each critical artifact.

training Activities that include specialized instruction and practice with the identified purpose of making one proficient in a skill or discipline.

trigger Indicator that a risk has occurred or is about to occur.

user The person or persons who operate or interact directly with the product. The user(s) and the customer(s) are often not the same person(s).

user support The assistance, training, and documentation a project provides to users of its software products in ensuring that the user is satisfied and able to use the product as intended. Typically, a support plan is drawn up to specify what will be, and what will not be, provided by the project team and for what period of time.

validation 1) Demonstrates that the product, as provided, fulfills its intended use. Validation assures “you built the right thing.” 2) The process of evaluating the mathematical formulation to ensure that it adequately describes the problem of interest, that is, that the computer simulation adequately represents the real world.

verification Addresses whether the work product properly reflects the specified requirements. Verification assures “you built it right.”

Acronyms

AQMC	ASC Quality Management Council
AR _x	Artifact (number)
ASCI	Accelerated Strategic Computing Initiative
ASC	Advanced Simulation and Computing
CM	Configuration Management
CMMI	Capability Maturity Model Integration®
CPR	Corporate Process Requirement
DOE	U.S. Department of Energy
DP	Defense Programs
LANL	Los Alamos National Laboratory
LLNL	Lawrence Livermore National Laboratory
LOF	level of formality
GP&G	<i>ASCI Software Quality Engineering: Goals, Principles, and Guidelines</i>
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
NNSA	National Nuclear Security Agency
PR _x	Practice (number)
PEM	Program Element Manager
PI	principal investigator
QC	Quality Criteria
S&CS/OC	Simulation and Computer Science/Ongoing Computing
SNL	Sandia National Laboratories
SCM	software configuration management
SNL	Sandia National Laboratories
SPG	supplemental practice guidance
SQE	Software Quality Engineering
SSP	Stockpile Stewardship Program
SSQAP	Sandia Software Quality Assurance Program
V&V	Verification and Validation

This page intentionally left blank.

APPENDIX A. LEVEL OF FORMALITY TABLES: TARGET RATINGS AND RULES OF THUMB

This appendix contains three tables pertaining to the level of formality (LOF) for a project. Table 4 contains the target ratings set by the AQMC for each practice and is organized by high, medium, and low LOF. Low LOF projects are expected to address a core of 10 practices, most of which correspond to the guidance provided in the “Specific Use Specification,” Sandia Software Quality Assurance Program (SS-R89727),²⁴ with target ratings of 3. Medium and High LOF projects are expected to address all 30 practices. Medium LOF projects primarily have target ratings of 3, and High LOF projects have target ratings of 4. Following is a brief description of level of practice implementation required for the three LOFs and their associated target ratings.

- **Low LOF = 3** The software project team has implemented 10 core practices. The team will likely have implemented other practices depending on the scope of its work and how far along it is with its software development activities.
- **Medium LOF = 3 (or 4 for PR17 and PR19)** The software project team has implemented all 30 practices as evidenced by documented processes and associated work products of significant content.
- **High LOF = 4** The software project team has implemented a final, not draft, process for conducting the practice and work products are in place supporting this practice. The team’s implementation of all 30 practices is considered complete.

The 10 practices required of Low LOF projects are foundational to project success. The target ratings are higher for PR17 and PR19 than for other practices under Medium LOF. These two practices are deemed more important by the ASC Program Office.

Table 5 provides a complete description for each of the target ratings 0 through 5 as well as information regarding the types of “evidence” (i.e., artifacts, information gleaned from interviews, etc.) expected for each rating level when independent appraisals are conducted using the *ASC Appraisal Method* described in Section 2.5.

Table 6 provides example “rules of thumb” that may help a project to define what level of implementation detail is necessary to meet the target ratings. These are general guidelines. For additional suggestions and support for ASC SQE practice implementation, please visit the ASC SQE website¹⁶. This website provides implementation support, including example “best practices,” recommended tools, and templates for implementing practices.

Table 4. AQMC Target Rating Level Expectations Based upon Determined LOF

Categories/Process Areas/Practices	Target Ratings		
	Low LOF	Medium LOF	High LOF
1. Integrated Teaming			
PR1. Document and maintain a strategic plan.	(a)	3	4
2. Graded Level of Formality			
PR2. Perform a risk-based assessment, determine level of formality and applicable practices, and obtain approvals.	3	3	4
3. Measurement and Analysis			
PR3. Document, monitor, and control lifecycle processes and their interdependencies, and obtain approvals.	(a)	3	4
PR4. Define, collect, and monitor appropriate process metrics.	(a)	3	4
PR5. Periodically evaluate quality issues and implement process improvements.	(a)	3	4
4. Requirements Development and Management			
PR6. Identify stakeholders and other requirements sources.	3	3	4
PR7. Gather and manage stakeholders' expectations, requirements, and constraints.	3	3	4
PR8. Derive, negotiate, manage, and trace requirements.	3	3	4
5. Risk Management			
PR9. Identify and analyze risk events.	3	3	4
PR10. Define, monitor, and implement the risk response.	3	3	4
6. Project Planning and Oversight			
PR11. Create and manage the project plan.	3	3	4
PR12. Track project performance versus project plan and implement needed (i.e., corrective) actions.	3	3	4
7. Technical Solution			
PR13. Communicate and review design.	(a)	3	4
PR14. Create required software and product documentation.	(a)	3	4
PR15. Identify and track third party software products and follow applicable agreements.	(a)	3	4
PR16. Identify, accept ownership, and manage assimilation of other software products.	(a)	3	4
8. Configuration Management			
PR17. Perform version control of identified software product artifacts.	3	4	4
PR18. Record and track issues associated with the software product.	(a)	3	4
PR19. Ensure backup and disaster recovery of software product artifacts.	3	4	4
9. Product Integration			
PR20. Plan and generate the release package.	(a)	3	4
PR21. Certify that the software product (i.e., code and its related artifacts) is ready for release and distribution.	(a)	3	4
10. Deployment and Lifecycle Support			
PR22. Distribute release to customers.	(a)	3	4
PR23. Define and implement a customer support plan.	(a)	3	4
PR24. Implement the training identified in the customer support plan.	(a)	3	4
PR25. Evaluate customer feedback to determine customer satisfaction.	(a)	3	4
11. Software Verification			
PR26. Develop and maintain a software verification plan.	(a)	3	4
PR27. Conduct tests to demonstrate that acceptance criteria are met and to ensure that previously tested capabilities continue to perform as expected.	(a)	3	4
PR28. Conduct independent technical reviews to evaluate adequacy with respect to requirements.	(a)	3	4
12. Training			
PR29. Determine project team training needed to fulfill assigned roles and responsibilities.	(a)	3	4
PR30. Track training undertaken by project team.	(a)	3	4
Total Target Rating	30	92	120

(a) Low LOF projects are expected to address a core of 10 practices, most of which correspond to the guidance provided in the "Specific Use Specification," Sandia Software Quality Assurance Program (SS-R89727),²⁴ with target ratings of 3. Low LOF projects are not required, but may choose, to address practices that have no target ratings.

Table 5. ASC SQE Appraisal Rating Scale

Rating	<p align="center">Rating Description</p> <p align="center">(NOTE: Descriptions below are provided to allow for further clarification on typical expectations at each rating level. These are not intended to be all-inclusive; thus projects may or may not exhibit all characteristics identified in the following descriptions.)</p>	Evidence
5	<p>Outstanding – the software project team has fully implemented this practice; meaning that a documented process exists for the practice, all team members are fully trained on the process, work products have been produced and managed, and practice plans and results have been shared with all appropriate stakeholders. The team’s implementation of the practice is a candidate for a best practice to be shared with other teams and/or institutionalized across multiple teams or the program.</p>	Both direct and indirect evidence are a strength
4	<p>Complete – the software project team has implemented a final (not draft) process for conducting the practice and work products are in place supporting this practice. Most project team members have been trained in the process implementation. Practice results have been shared with appropriate stakeholders.</p>	Direct evidence is a strength, indirect evidence is OK
3	<p>Good – the software project team has implemented this practice; however, there are still a few activities that need to be addressed (e.g., training, finalizing work products, etc.). For example, a draft of the process for conducting the practice exists or a completed documented process exists with most of the team (but not all) complying with the process. The team has made significant progress in rolling-out an implementation for the process, and draft work products that contain significant content exist.</p>	Direct evidence is OK, indirect evidence is a strength, OK, or weak
2	<p>Fair – the software project team has a preliminary process (e.g., a detailed outline, a well-understood ad hoc team process that is not documented, etc.) for implementing this practice. There may be a preliminary plan about how to proceed with the process, and implementation and preliminary work products exist.</p>	Direct evidence is weak, indirect evidence is a strength, OK, or weak
1	<p>Limited –the software project team has proposed that this practice be implemented and activities and resources for the practice are in the planning stages. It is evident that the project is committed to implementing this practice. At this level, it is typical that resources have not yet been allocated for fulfillment of the practice.</p>	Direct evidence is non-existent, indirect evidence is a strength, OK, or weak
0	<p>Absent – the software project team has not yet addressed the implementation of this practice.</p>	Both direct and indirect evidence are non-existent
NA	<p>Not Applicable – the software project team has determined that this practice is not applicable to its code development environment. A value of NA must be accompanied by an explanation from the team describing why the practice will not be followed.</p>	NA

Table 6. Rules of Thumb for LOF

	Low Formality	Medium Formality	High Formality
Artifacts	Artifacts contain key details and may take the form of notes in an engineering notebook, hardcopy of drawing notes on a whiteboard, meeting notes, presentation materials, and email. Artifacts are available to at least the artifact owner and principal investigator (PI).	Artifacts contain significant detail, including key concepts and are likely in draft form. Artifacts are identified in the project plan and are stored in a repository available to all project team members.	Artifacts are complete and in final form. Artifacts are identified in the project plan. Format of the artifacts includes formal project, product, or process reports, or memos stored in a repository available to all project stakeholders.
Reviews	Takes the form of meeting notes, emails, and paired programming practices. Reviews are witnessed/approved as needed. Reviews consist of at least one reviewer who is knowledgeable and independent of artifact construction. Review records become artifacts.	Low formality plus PI and appropriate management are involved in reviews. Customers are informed of status of reviews. Key concepts of artifacts are reviewed and approved by team members and appropriate management. Review records become artifacts.	Reviews are scheduled in the project plan. Attendees include management, PI, project team, subject matter experts and/or key stakeholders. Review results require approvals by appropriate management and stakeholders. Findings and issues are maintained in a formal report or issue tracking system. Review records become artifacts.
Training	Takes the form of mentoring and self-paced training, including reading books, journals, seminars, and self-study training material. Training records may include e-mail acknowledgement to team lead or PI. Team maintains a record of skills and training required to develop the skill set. Training records become artifacts.	Low formality plus identification of critical skills redundancy (where cross-training results in several team members who are knowledgeable of key areas). Feedback on effectiveness of training experiences is collected. Training records become artifacts.	Medium formality plus gathering of metrics for gauging effectiveness of training are identified, collected and applied. Training format includes to university and college degree programs, professional certifications, on and off-site classroom training, and computer-based training. Training records become artifacts.
Tools	Generic tools such as manual notebooks, calculators or common desktop tools such as office automation (word processing, spreadsheet, presentation, e-mail, project management). Key project members have access to these tools.	Low formality plus tools of a more specialized nature to address specific tasks (for example, software tools such as DOORS for requirements management and SourceForge for collaborative environments). Tools are available to appropriate project members and appropriate management and stakeholders.	Medium formality plus all appropriate management and stakeholders have access. Ideally, selected tools are a program or corporate resource.

APPENDIX B. MAPPING TO CORPORATE AND FEDERAL SQE REQUIREMENTS

The ability to easily trace the Software Quality Plan to both federal and Corporate SQE requirements is important not only to provide evidence of achievement of these requirements, but also to provide a better understanding of the drivers behind the ASC SQE practices. The following tables provide mappings to key drivers to meet this purpose:

- Table 7, below, provides the mapping of the Software Quality Plan ASC SQE practices to both the process areas and the dimensions required in the SNL CPR001.3.6¹¹. In addition, the ASC SQE practices are mapped to the Sandia National Laboratories, Software Quality Assurance Program (SSQAP), an implementation of CPR001.3.6, which addresses compliance with DOE Order 414.1C and NNSA-QC-1.
- Table 1 in the “Mapping ASC Software Quality Plan to Customer Directives” document²³ provides the mapping of the Software Quality Plan to the DOE/NNSA ASCI Software Quality Engineering GP&G¹⁵.
- Table 2 in the “Mapping ASC Software Quality Plan to Customer Directives” document²³ provides the mapping of the Software Quality Plan to the DOE Quality Criteria, Revision 10 (QC-1)¹³.

Table 7. Mapping to CPR001.3.6

CPR001.3.6 Process Area and Dimension Mappings		
CPR001.3.6 Process Area	SSQAP Practice-Level Tier ²⁴	Software Quality Plan SQE Practices
Project Planning and Oversight		PR11, R12
<ul style="list-style-type: none"> • Estimate the Scope of the Project , Estimates of Work Product and Task Attributes, Budget and Schedule, and the Project Plan • Define Project Life Cycle • Determine Estimates of Effort and Cost • Plan Needed Acquisitions and Suppliers • Obtain Plan Commitment • Plan for Project Resources, Needed Knowledge and Skills, Stakeholder Involvement, and Data Management Review Plans That Affect the Project • Identify Project Risks • Reconcile Work and Resource Levels • Monitor Project Planning Parameters, Commitments, Data Management, and Selected Supplier Processes • Monitor Stakeholder Involvement and Project Risks • Conduct Progress and Milestone Reviews 	<ul style="list-style-type: none"> P1 P1 P1 P1 P2 P2 P2 P2 P3 P3 	<ul style="list-style-type: none"> PR11 PR3 PR11 PR6, PR11, PR15, PR16 PR11 PR11, PR26, PR29 PR9 PR12 PR12, PR15, PR17, PR19 PR10, PR12 PR12
Risk Management		PR9, PR10
<ul style="list-style-type: none"> • Determine Risk Sources and Categories • Define Risk Parameters • Establish a Risk Management Strategy • Identify Risks • Evaluate, Categorize, and Prioritize Risks • Develop Risk Mitigation Plans • Implement Risk Mitigation Plans • <i>Safety Software, Graded LOF with Risk-based Approach (ASC-specific)</i> 	<ul style="list-style-type: none"> P1 P1 P1 P1 P1 P1 P1 NA 	<ul style="list-style-type: none"> PR2, PR9 PR2, PR9 PR2, PR10 PR2, PR9 PR2, PR9 PR2, PR10 PR2, PR10 PR2

Table 7. Mapping to CPR001.3.6 (continued)

CPR001.3.6 Process Area Mappings		
CPR001.3.6 Process Area	SSQAP Practice-Level Tier ²⁴	Software Quality Plan SQE Practices
Requirements Development and Management		PR7, PR8
<ul style="list-style-type: none"> Obtain an Understanding of and Commitment to Requirements Manage Requirements Changes Maintain Bidirectional Traceability of Requirements Identify Inconsistencies Between Project Work and Requirements Elicit Needs Develop the Customer Requirements Analyze Requirements Establish Product and Product Component Requirements Allocate Product Component Requirements Identify Interface Requirements Establish Operational Concepts and Scenarios, and a Definition of Required Functionality Analyze Requirements to Achieve Balance Validate Requirements 	<ul style="list-style-type: none"> P1 P1 P2 P2 P1 P1 P1 P1 P2 P2 P2 P2 P2 	<ul style="list-style-type: none"> PR7, PR8 PR8 PR8 PR7, PR8, PR13, PR14 PR6, PR7 PR7 PR8 PR8 PR8 PR8, PR13 PR13 PR8 PR28
Technical Solution		PR13, PR14, PR15, PR16
<ul style="list-style-type: none"> Develop Alternative Solutions and Selection Criteria Select Product Component Solutions Design the Product or Product Component 	<ul style="list-style-type: none"> P2 P2 P2 	<ul style="list-style-type: none"> PR13 PR13 PR13
Verification & Validation		PR26, PR27, PR28
Verification		PR26
<ul style="list-style-type: none"> Select Work Products for Verification Establish the Verification Environment, Procedures and Criteria Prepare for Peer Reviews Conduct Peer Reviews Analyze Peer Review Data Perform Verification Analyze Verification Results 	<ul style="list-style-type: none"> P2 P2 P2 P3 P3 P3 P3 	<ul style="list-style-type: none"> PR20, PR26 PR28 PR28 PR28 PR27, PR28 PR27, PR28
Validation		
<i>The following are the responsibility of the ASC V&V Program or the 'specific use' user:</i>		
<ul style="list-style-type: none"> Select Products for Validation Establish the Validation Environment and Validation Procedures and Criteria Perform Validation Analyze Validation Results Evaluate Selected Supplier Work Products Accept the Acquired Product 	<ul style="list-style-type: none"> P2 P2 P3 P3 P3 P3 	

Table 7. Mapping to CPR001.3.6 (continued)

CPR001.3.6 Process Area Mappings		
CPR001.3.6 Process Area	SSQAP Practice-Level Tier ²⁴	Software Quality Plan SQE Practices
Deployment and Lifecycle Support		PR22, PR23, PR24
<ul style="list-style-type: none"> Assemble Product Components Package and Deliver the Product or Product Component Transition Supplier Products Establish a Technical Data Package for Each Release Develop Product Support Concept and Documentation 	<p>P2</p> <p>P2</p> <p>P2</p> <p>P2</p> <p>P3</p>	<p>PR14, PR16, PR20</p> <p>PR21, PR22</p> <p>PR15, PR16, PR20, PR21, PR22</p> <p>PR20, PR23</p> <p>PR14, PR19, PR23, PR24, PR25</p>
Configuration Management		PR17, R18, PR19
<ul style="list-style-type: none"> Identify Configuration Items Establish a Configuration Management System Create or Release Baselines Control Configuration Items Establish Configuration Management Records Track Change Requests Perform Configuration Audits 	<p>P1</p> <p>P1</p> <p>P1</p> <p>P1</p> <p>P2</p> <p>P2</p> <p>P2</p>	<p>PR17, PR20</p> <p>PR17</p> <p>PR17</p> <p>PR17</p> <p>PR18, PR19</p> <p>PR18</p> <p>Section 4.2.2 Overview, PR17, PR21</p>
Measurement and Analysis		PR3, PR4, PR25
<ul style="list-style-type: none"> Establish Measurement Objectives Specify Measures, Data Collection and Storage Procedures, and Analysis Procedures Collect Measurement Data Analyze Measurement Data Store Data and Results Communicate Results <i>Self-Independent, External Appraisals (ASC-specific)</i> 	<p>P2</p> <p>P2</p> <p>P2</p> <p>P3</p> <p>P3</p> <p>P3</p>	<p>PR4, PR12, PR25</p> <p>PR4, PR12</p> <p>PR4, PR12</p> <p>PR4, PR5, PR12</p> <p>PR4, PR5, PR12</p> <p>PR5, PR12</p> <p>PR5, Section 5.0</p>
Integrated Product and Teaming		PR20, PR21, PR1
<ul style="list-style-type: none"> Determine Integration Sequence Establish the Product Integration Environment, Procedures and Criteria Review Interface Descriptions for Completeness Manage Interfaces Confirm Readiness of Product Components for Integration Evaluate Assembled Product Components Establish the Project's Shared Vision, Integrated Teams, and Team Structure Allocate Requirements to Integrated Teams Ensure Collaboration among Interfacing Teams 	<p>P2</p> <p>P2</p> <p>P2</p> <p>P3</p> <p>P3</p> <p>P3</p> <p>P3</p> <p>P3</p> <p>P3</p>	<p>PR20</p> <p>PR20</p> <p>PR13, PR21</p> <p>PR20</p> <p>PR21</p> <p>PR21</p> <p>PR1</p> <p>PR1, PR6, PR7</p> <p>PR1, PR6, PR7</p>

Table 7. Mapping to CPR001.3.6 (completed)

CPR001.3.6 Dimension Mappings		
CPR001.3.6 Dimension	SSQAP Practice-Level Tier ²⁴	Software Quality Plan Key SQE Practices
Identified and involved stakeholders		PR6
<ul style="list-style-type: none"> • Manage Stakeholder Involvement • Manage Dependencies • Resolve Coordination Issues 	P2 P2 P2	PR6 PR7 PR7
Ongoing process monitoring and control	<i>There is no comparable practice</i>	PR4, PR5
Collected improvement information	<i>There is no comparable practice</i>	PR4, PR5, PR25
Objective evaluations		PR5
<ul style="list-style-type: none"> • Objectively Evaluate Processes, Work Products, and Services • Communicate and Ensure Resolution of Noncompliance Issues • Establish Records 	P4 P4 P4	PR3, PR5, PR25 PR5 PR4, PR18, PR24, PR25
Quantitative objectives defined for processes		PR4
<ul style="list-style-type: none"> • Select Processes • Establish Process-Performance Measures • Establish Quality and Process-Performance Objectives, Process-Performance Baselines, and Process-Performance Models 	P3 P3 P4	PR4 PR4 PR4
Stable subprocess performance	<i>There is no comparable practice</i>	PR4
Training		PR29, PR30
<ul style="list-style-type: none"> • Establish the Strategic Training Needs • Determine Which Training Needs Are the Responsibility of the Organization • Establish Training Records • Deliver Training • Assess Training Effectiveness • Establish an Organizational Training Tactical Plan & Training Capability 	P1 P1 P1 P2 P2 P3	PR29 Section 4.4.1 Overview, PR29 PR30 PR30 Section 4.4.1 Overview, PR30 Section 4.4.1 Overview
Problem reporting and corrective action		PR5, PR18
<ul style="list-style-type: none"> • Analyze Issues • Take Corrective Action • Manage Corrective Action 	P2 P2 P2	PR5, PR18, PR25, PR28 PR4, PR5, PR12 PR18

APPENDIX C. MAPPING TO VERSION 2.0 OF THE SOFTWARE QUALITY PLAN

The following general changes were made in Version 3.0 of the Software Quality Plan:

- Rename practice areas to use the terms (i.e., process areas and dimensions) from CPR001.3.6
- Broaden the scope of the practices to fully encompass CPR001.3.6 directives
- Address safety software to the extent necessary
- Include a section (i.e., Section 2) that explains the overall SNL ASC SQE approach
- Include a section (i.e., Section 3) that explains the organization, terms, and conventions used in the document
- Collapse the two-part Software Quality Plan into one document that includes mappings

These changes resulted in only four primary changes regarding the ASC SQE practices (see chart below for references to the process areas): Version 2.0 process areas 7 and 8 were combined into Version 3.0 process area 7; PR22 was moved into process area 10; the scope of some practices and artifacts were revised; and one new artifact (AR27) was established. Table 8 maps the revisions in Version 3.0 to Version 2.0 of the Software Quality Plan.

Table 8. Mapping to Previous Version of the Software Quality Plan

Category	Version 3.0 12 Process Areas and 30 Practices		Version 2.0 13 Practice Areas and 30 Practices	
Project Management	1. Integrated Teaming	PR1	1. Strategic Planning	PR1
	2. Graded Level of Formality	PR2	2. Determination of Applicable Practices and LOF	PR2
	3. Measurement and Analysis	PR3, PR4, PR5	3. Process Implementation and Improvement	PR3, PR4, PR5
	4. Requirements Development & Management	PR6, PR7, PR8	4. Requirements Engineering	PR6, PR7, PR8
	5. Risk Management	PR9, PR10	5. Risk Management	PR9, PR10
	6. Project Planning & Oversight	PR11, PR12	6. Project Planning, Tracking & Oversight	PR11, PR12
Software Engineering	7. Technical Solution	PR13, PR14, PR15, PR16	7. Software Development	PR13, PR14, PR15, PR16
	8. Configuration Management	PR17, PR18, PR19	8. Integration of 3rd Party or Other Software	PR17, PR18, PR19
	9. Product Integration	PR20, PR21	9. Configuration Management	PR17, PR18, PR19
	10. Deployment & Lifecycle Support	PR22, PR23, PR24, PR25	10. Release & Distribution Management	PR20, PR21, PR22
			11. Customer Support	PR23, PR24, PR25

Table 8. Mapping to Previous Version of the Software Quality Plan (continued)

Category	Version 3.0 12 Process Areas and 30 Practices		Version 2.0 13 Practice Areas and 30 Practices	
Software Verification	11. Software Verification & Validation	PR26, PR27, PR28	12. Software Verification	PR26, PR27, PR28
Training	12. Training	PR29, PR30	13. Training	PR29, PR30

DISTRIBUTION

External Distribution:

U.S. Department of Energy
1000 Independence Avenue SW
Washington, DC 20585
Attn: Robert Meisner, DP-51 (1)
Karen Pao, DP-51 (1)

Los Alamos National Laboratory
Mail Station 5000
P.O. Box 1663
Los Alamos, NM 87545
Attn: Mark Anderson
Verification & Validation, PEM

University of California
Lawrence Livermore National Laboratory
7000 East Avenue
P.O. Box 808
Livermore, CA 94550
Attn: Joseph Sefcik

SNL Distribution: Electronic Only

Name/Organization	Name/Organization	Name/Organization
Thomas Bickel, 1200	Ann Hodges, 5212	Martin Pilch, 1221
Steve Bova, 1541	Ed Hoffman, 8964	Arthur Ratzel, 1500
Russell DePriest, 1384	David Karelitz, 9326	Garth Reese, 1542
Karen Devine, 1416	Philip Kegelmeyer, 8962	Allen Robinson, 1431
Stefan Domino, 1541	Eric Keiter, 1437	Chris Northrup, 1055
Sudip S. Dosanjh, 1420	Sue Kelly, 1423	Bill Scherzinger, 1524
Richard Drake, 1431	Ron Kensek, 1341	Joseph Schofield, 9501
Clif Drumm, 1341	Bob Kerr, 1421	Peter Schultz, 1435
Mike Eldred, 1411	Mark Kiefer, 1652	Judy Sturtevant, 9328
Ernest Friedman-Hill, 8962	Chris Lamb, 9512	Timothy G. Trucano, 1411
Anthony Giunta, 1544	Randy Lober, 1542	Jennifer Turgeon, 6325
Michael Glass, 1541	Molly Minana, 6325	Larry Walker, 2900
John Greenfield, 9326	Ken Moreland, 1424	Lee Ward, 1422
Arne Gullerud, 1524	Harold S. Morgan, 1030	Bob Whiteside, 8964
Patricia Hackney, 6325	Pat Notz, 1541	Alan Williams, 1541
Jason Hales, 1542	Tolulope Okusanya, 1541	Michael Williamson, 48271
James Handrock, 8200	Steven Owen, 1421	Steve Wix, 1734
Karen Haskell, 9326	Constantine Pavlakos, 9326	David E. Womble, 1540
Michael Heroux, 1416	James Peery, 1400	Alan Wright, 1112
Howard H. Hirano, 8960	Kendall Pierson, 1542	Paul Yarrington, 1220

Housekeeping Distribution:

1 S0899 Technical Library, 9536 (electronic copy)

This page intentionally left blank.

