NOT MEASUREMENT SENSITIVE

DOE G 414.1-4
Approved 6-17-05
Certified 11-3-10

SAFETY SOFTWARE GUIDE
for USE with
10 CFR 830 Subpart A, *Quality Assurance Requirements, and DOE O 414.1C, Quality Assurance*

[This Guide describes suggested nonmandatory approaches for meeting requirements. Guides are not requirements documents and are not construed as requirements in any audit or appraisal for compliance with the parent Policy, Order, Notice, or Manual.]

U.S. DEPARTMENT OF ENERGY
Washington, D.C.

DISTRIBUTION:
http://www.directives.doe.gov

INITIATED BY:
Office of Environment, Safety and Health

FOREWORD

This Department of Energy (DOE) Guide is approved by the Office of Environment, Safety and Health and is available for use by all DOE and National Nuclear Security Administration (NNSA) elements and their contractors. This Guide revises and supersedes earlier guidance identified in Appendix B to include new and updated information.

Comments, including recommendations for additions, modifications, or deletions, and other pertinent information, should be sent to the following.

Gustave E. Danielson, Jr.
U.S. DOE
Office of Quality Assurance Programs
1000 Independence Avenue SW
EH-31/270CC
Washington, DC 20585-0270
Phone: 301-903-2954
Fax: 301-903-4120
E-mail: bud.danielson@hq.doe.gov

Richard H. Lagdon, Jr. (Chip)
U.S DOE
Director, Office of Quality Assurance Programs
1000 Independence Avenue SW
EH-31/270CC
Washington, DC 20585-0270
Phone: 301-903-4218
Fax: 301-903-4120
E-mail: chip.lagdon@eh.doe.gov

Guides are part of the DOE directives system and are used to provide supplemental information regarding DOE/NNSA expectations for fulfilling requirements contained in Policies, Rules, Orders, Manuals, Notices, and Regulatory Standards. Guides are also used to identify Government and non-Government standards and acceptable methods for implementing DOE/NNSA requirements. Guides are not substitutes for requirements nor do they introduce new requirements or replace technical standards used to describe established practices and procedures.

CONTENTS

CONTENTS (continued)

BACKGROUND

The use of digital computers and programmable electronic logic systems has increased significantly since 1995, and their use is evident in safety applications at nuclear facilities across the Department of Energy (DOE or Department) complex. The commercial industry has increased attention to quality assurance of safety software to ensure that safety systems and structures are properly designed and operate correctly. Recent DOE experience with safety software has led to increased attention to the safety-related decision making process, the quality of the software used to design or develop safety-related controls, and the proficiency of personnel using the safety software.

The Department has recognized the need to establish rigorous and effective requirements for the application of quality assurance (QA) programs to safety software. In evaluating Defense Nuclear Facilities Safety Board (DNFSB) recommendation 2002-1 and through assessing the current state of safety software, the Department concluded that an integrated and effective Software Quality Assurance (SQA) infrastructure must be in place throughout the Department's nuclear facilities. This is being accomplished through the Implementation Plan for Defense Nuclear Facilities Safety Board Recommendation 2002-1, *Quality Assurance for Safety Software at Department of Energy Defense Nuclear Facilities*.

To ensure the quality and integrity of safety software, DOE directives are being developed and revised based on existing SQA industry or Federal agency standards This resulted in the development and issuance of DOE O 414.1C, *Quality Assurance*, dated 6-17-05, which includes specific SQA requirements, this Guide and the DOE Standard 1172-2003, Safety Software Quality Assurance Functional Area Qualification Standard, dated December 2003. The SQA requirements are to be implemented by DOE and its contractors. Nuclear facility contractors must implement the SQA requirements under their QA program for 10 CFR 830, Subpart A, Quality Assurance Requirements. Thus, the intent of this Guide is to provide instructional guidance for application of DOE O 414.1C safety software requirements.

1. INTRODUCTION

1.1 PURPOSE

This Department of Energy (DOE or Department) Guide provides information plus acceptable methods for implementing the safety software quality assurance (SQA) requirements of DOE O 414.1C, *Quality Assurance,* dated 6-17-05. DOE O 414.1C requirements supplement the quality assurance program (QAP) requirements of Title 10 Code of Federal Regulations (CFR) 830, Subpart A, Quality Assurance, for DOE nuclear facilities and activities. The safety SQA requirements for DOE, including the National Nuclear Security Administration (NNSA), and its contractors are necessary to implement effective quality assurance (QA) processes and achieve safe nuclear facility operations.

DOE promulgated the safety software requirements and this guidance to control or eliminate the hazards and associated postulated accidents posed by nuclear operations, including radiological operations. Safety software failures or unintended output can lead to unexpected system or equipment failures and undue risks to the DOE/NNSA mission, the environment, the public, and the workers. Thus DOE G 414.1-4 has been developed to provide guidance on establishing and implementing effective QA processes tied specifically to nuclear facility safety software applications. DOE also has guidance[1] for the overarching QA program, which includes safety software within its scope. This Guide includes software application practices covered by appropriate national and international consensus standards and various processes currently in use at DOE facilities.[2] This guidance is also considered to be of sufficient rigor and depth to ensure acceptable reliability of safety software at DOE nuclear facilities.

This guidance should be used by organizations to help determine and support the steps necessary to address possible design or functional implementation deficiencies that might exist and to reduce operational hazards-related risks to an acceptable level. Attributes such as the facility life-cycle stage and the hazardous nature of each facility's operations should be considered when using this Guide. Alternative methods to those described in this Guide may be used provided they result in compliance with the requirements of 10 CFR 830 Subpart A and DOE O 414.1C. Another objective of this guidance is to encourage robust software quality methods to enable the development of high quality safety applications.

1.2 SCOPE

This Guide is intended for use by all DOE/NNSA organizations and their contractors to assist in developing site and facility specific safety SQA processes and procedures compliant with 10 CFR 830 Subpart A and DOE O 414.1C.

The Department's objectives for safety software requirements include—

[1] DOE G 414.1-2, *Quality Assurance Management System Guide for use with 10 CFR 830.120 and DOE O 414.1,* dated 6-17-99.

[2] See Appendix G for list of consensus standards.

- grading SQA requirements based on risk, safety, facility life-cycle, complexity, and project quality requirements;
- applying SQA requirements to software life-cycle phases;
- developing procurement controls for acquisition of computer software and hardware that are provided with supplier-developed software and/or firmware;
- documenting and tracking customer requirements;
- managing software configuration throughout the life-cycle phases;
- performing verification and validation (V&V)[3] processes;
- performing reviews of software configuration items, including reviewing the safety implications identified in the failure analysis and fault tolerance design; and
- training personnel who use and apply software in safety applications.

The scope of this Guide includes software applications that meet safety software definitions as stated in DOE O 414.1C. This includes software applications important to safety that may be included or associated with structures, systems, or components (SSCs) for less than hazard category 3 facilities. *Safety Software* includes safety system software, safety and hazard analysis software and design software, and safety management and administrative control software.

Safety system software is software for a nuclear facility[4] that performs a safety function as part of an SSC and is cited in either (1) a DOE approved documented safety analysis or (2) an approved hazard analysis per DOE P 450.4 *Safety Management System Policy,* dated 10-15-96, and the DEAR clause.

Safety and hazard analysis software and design software is software that is used to classify, design, or analyze nuclear facilities. This software is not part of an SSC but helps to ensure the proper accident or hazards analysis of nuclear facilities or an SSC that performs a safety function.

Safety management and administrative controls software is software that performs a hazard control function in support of nuclear facility or radiological safety management programs or Technical Safety Requirements or other software that performs a control function necessary to provide adequate protection from nuclear facility or radiological hazards. This software supports eliminating, limiting, or mitigating nuclear hazards to workers, the public, or the environment as addressed in 10 CFR 830, 10 CFR 835, and the DEAR ISMS clause.

Additional definitions are included in Appendix A, Acronyms and Definitions.

[3] Verification and validation in this Guide includes ASME's NQA-1 terms design verification and acceptance testing.

[4] Per 10 CFR 830, quality assurance requirements apply to all DOE nuclear facilities including radiological facilities (see 10 CFR 830, DOE Std 1120, and the DEAR clause).

Although this Guide has been developed for DOE nuclear facility software, it may also be useful for ensuring the quality of other software important to mission critical functions, environmental protection, health and safety protection, safeguards and security, emergency management, or assets protection.

1.3 RESPONSIBILITY FOR SAFETY SOFTWARE

The Assistant Secretary for Environment, Safety and Health has the lead responsibility for promulgating requirements and guidance through the directives system for safety software per DOE O 414.1C. The organizations that use software should determine whether to qualify the software for safety applications. Organizations should coordinate SQA procedures with their respective Chief Information Officers and other appropriate organizations. DOE line organizations are responsible for providing direction and oversight of the contractor implementation of SQA requirements.

1.4 SAFETY SOFTWARE QUALITY ASSURANCE

The scope of the Department's QA Rule, 10 CFR 830 Subpart A, is stated as "This subpart establishes quality assurance requirements for contractors conducting activities, including providing items or services, that affect, or may affect, nuclear safety of DOE nuclear facilities." The scope of the QA Rule encompasses the contractor's conduct of activities as they relate to safety software (items or services). Therefore the contractor's QAP includes safety software within its scope. DOE O 414.1C establishes the safety software QA requirements to be implemented under the Rule. 10 CFR 830 Subpart A and DOE O 414.1C require contractors to perform safety software work in accordance with the applicable criteria.

The various sections of this Guide discuss the application of the QA criteria from DOE O 414.1C and 10 CFR 830 Subpart A to the ten SQA work activities. Table 1 provides an illustration of how the SQA work activities satisfy the QA criteria.

1.5 SOFTWARE QUALITY ASSURANCE PROGRAM

It is important that SQA is part of an overall QAP required for nuclear facilities in accordance with 10 CFR 830 Subpart A and DOE O 414.1C. Regardless of the application of the software, an appropriate level of quality infrastructure should be established and a commitment made to maintain this infrastructure for the safety software.

An SQA program establishes the appropriate safety software life-cycle practices, including safety design concepts, to ensure that software functions reliably and correctly performs the intended work specified for that safety software. In other words, SQA's role is to minimize or prevent the failure of safety software and any undesired consequences of that failure. The rigor imposed by SQA is driven by the intended use of the safety software. More importantly, the rigor of SQA should address the risk of use of such software. Effective safety software quality is one method for avoiding, minimizing, or mitigating the risk associated with the use of the software.

Table 1. An Illustration of Quality Assurance (QA) Criteria (10 CFR 830 Subpart A & DOE O 414.1C) Applicability to Software Quality Assurance (SQA) Work Activities

SQA Work Activities / 10 CFR 830 QA Criteria and DOE O 414.1C	Software (sw) project management and quality planning	Sw risk mgmt	Sw configuration mgmt	Procurement & supplier mgmt	Sw rqmts identification & mgmt	Sw design & implementation	Sw safety	Verification & validation	Prblm rpting & corrective action	Training of … safety sw
Program	X	X	X	X	X	X	X	X	X	X
Training & Qualification										X
Quality Improvement								X	X	
Documents and Records	X	X	X	X	X	X	X	X	X	X
Work Processes	X	X	X	X	X	X	X	X	X	X
Design			X		X	X	X			
Procurement				X						
Inspection & Acceptance Testing								X		
Management Assessment	X		X	X	X	X	X	X	X	X
Independent Assessment	X	X	X	X	X	X	X	X	X	X

Note: This table is only an illustration of QA criteria applicability. Actual application will be described in the organization's QA program and safety software work process documents. For example, an independent assessment may be performed on any safety software quality element.

The goal of SQA for safety system software is to apply the appropriate quality practices to ensure the software performs its intended function and to mitigate the risk of failure of safety systems to acceptable and manageable levels. SQA practices are defined in national and international consensus standards. SQA cannot address the risks created by the failure of other system components (hardware, data, human process, power system failures) but can address the software "reaction" to effects caused by these types of failures. SQA should not be isolated from system level QA and other system level activities. In many instances, hardware fail-safe methods are implemented to mitigate risk of safety software failure. Additionally other interfaces such as hardware and human interfaces with safety software should implement QA activities.

2. SAFETY SOFTWARE TYPES AND GRADING

2.1 SOFTWARE TYPES

Software typically is either custom developed or acquired software. Further characterizing these two basic types aids in the selection of the applicable practices and approaches for the SQA work activities. For the purposes of this Guide, five types of software have been identified as commonly used in DOE applications: (1) custom developed, (2) configurable, (3) acquired, (4) utility calculation, and (5) commercial design and analysis.

Developed and acquired software types as discussed in American Society of Mechanical Engineers (ASME) NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications* are compatible with these five software types. Developed software as described in ASME NQA-1-2000 is directly associated with custom developed, configurable, and utility calculation software. Acquired software included in this Guide is easily mapped to that of acquired software in ASME NQA-1-2000. ASME NQA-1-2000 uses acquired and procured software terms interchangeably.[5] This Guide includes an additional software type of commercial design and analysis software that is not directly related to either developed or acquired software. Safety software quality requirements can only be specified through work activities described in contractual agreements with the supplier of the facility design and analysis services.

Custom developed software is built specifically for a DOE application or to support the same function for a related government organization. It may be developed by DOE or one of its management and operating (M&O) contractors or contracted with a qualified software company through the procurement process. Examples of custom developed software includes material inventory and tracking database applications, accident consequence applications, control system applications, and embedded custom developed software that controls a hardware device.

Configurable software is commercially available software or firmware that allows the user to modify the structure and functioning of the software in a limited way to suit user needs. An example is software associated with PLCs.

[5] ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Subpart* 2.7 Section 300, American Society of Mechanical Engineers, New York, New York, 2001, p. 105.

Acquired software is generally supplied through basic procurements, two-party agreements, or other contractual arrangements. Acquired software includes commercial off-the-shelf (COTS) software, such as operating systems, database management systems, compilers, software development tools, and commercial calculational software and spreadsheet tools (e.g., Mathsoft's MathCad and Microsoft's Excel). Downloadable software that is available at no cost to the user (referred to as freeware) is also considered acquired software. Firmware is acquired software. Firmware is usually provided by a hardware supplier through the procurement process and cannot be modified after receipt.

Utility calculation software typically uses COTS spreadsheet applications as a foundation and user developed algorithms or data structures to create simple software products. The utility calculation software within the scope of this document is used frequently to perform calculations associated with the design of an SSC. Utility software that is used with high frequency may be labeled as custom software and may justify the same safety SQA work activities as custom developed software.[6] With utility calculation software, it is important to recognize the difference between QA of the algorithms, macros, and logic that perform the calculations versus QA of the COTS software itself. Utility calculation software includes the associated data sets, configuration information, and test cases for validation and/or calibration.

Commercial design and analysis software is used in conjunction with design and analysis services provided to DOE from a commercial contractor. An example would be where DOE or an M&O contractor contracts for specified design services support. The design service provider uses its independently developed or acquired software without DOE involvement or support. DOE then receives a completed design. Procurement contracts can be enhanced to require that the software used in the design or analysis services meet the requirements in DOE O 414.1C.

2.2 GRADED APPLICATION

Proper implementation of DOE O 414.1C will be enhanced by grading safety software based on its application. Safety software grading levels should be described in terms of safety consequence and regulatory compliance. This Guide utilizes the grading levels and the software types (custom developed, configurable, acquired, utility calculations, and commercial design and analysis tools) to recommend how the SQA work activities are applied. The grading levels are defined as follows.

Level A: This grading level includes safety software applications that meet one or more of the following criteria.

1. Software failure that could compromise a limiting condition for operation.
2. Software failure that could cause a reduction in the safety margin for a safety SSC that is cited in DOE approved documented safety analysis.
3. Software failure that could cause a reduction in the safety margin for other systems such as toxic or chemical protection systems that are cited in either (a) a DOE approved

[6] ASME NQA-1-2000, op.cit., Part 4, Subpart 4.1, Section 101.1, p. 227.

documented safety analysis or (b) an approved hazard analysis per DOE P 450.1 and the DEAR ISMS clause.

4. Software failure that could result in nonconservative safety analysis, design, or misclassification of facilities or SSCs.

Level B: This grading level includes safety software applications that do not meet Level A criteria but meet one or more of the following criteria.

1. Safety management databases used to aid in decision making whose failure could impact safety SSC operation.

2. Software failure that could result in incorrect analysis, design, monitoring, alarming, or recording of hazardous exposures to workers or the public.

3. Software failure that could comprise the defense in depth capability for the nuclear facility.

Level C: This grading level includes software applications that do not meet Level B criteria but meet one or more of the following criteria.

1. Software failure that could cause a potential violation of regulatory permitting requirements.

2. Software failure that could affect environment, safety, health monitoring or alarming systems.

3. Software failure that could affect the safe operation of an SSC.

The grading level criteria should provide for a higher grade level for software in nuclear facilities categorized as Category 1, 2 or 3 and the lower grading level for software in facilities categorized as less than Category 3. Table 2 illustrates the association of grading criteria described above to facility categorization.

Using the grading levels and the safety software types in Table 2, select and implement applicable software quality work activities from the following list to ensure that safety software performs its intended functions. DOE O 414.1C specifies that ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications*, or other national or international consensus standards that provide an equivalent level of quality assurance requirements as ASME NQA-1-2000 must be used. As specified in DOE O 414.1C, the standards used must be specified by the user and approved by DOE. This Guide provides acceptable implementation strategies and appropriate standards for these work activities.

1. Software project management and quality planning.

2. Software risk management.

3. Software configuration management (SCM).

4. Procurement and supplier management.
5. Software requirements identification and management.
6. Software design and implementation.
7. Software safety.
8. V&V.
9. Problem reporting and corrective action.
10. Training of personnel in the design, development, use, and evaluation of safety software.

Table 2. Grading Criteria and Facility Categorization Illustration

	Nuclear Facilities 1,2 3			Nuclear Facilities <3		
Safety Software	**A**	**B**	**C**	**A**	**B**	**C**
Safety System Software	**X**	**X**				**X**
Safety & Hazard Analysis Software & Design Software*	**X**	**X**	**X**	**X**	**X**	**X**
Safety Management & Admin Controls Software	**X**	**X**	**X**			**X**

*Safety and hazard analysis software and design software includes software used to classify facilities. Because this software is used before the facility classification determination, the safety and hazard analysis software and design software type has been identified as being applicable for all grading levels in all categories of facilities.

The determination of what constitutes safety software is made by the organization applying the software based upon the requirements in DOE O 414.1C, and 10 CFR 830 Subparts A and B. The application of the software determines whether it is safety related and how it should be graded. Therefore, the organization applying the software is responsible for evaluating and designating the software as safety software and then ensuring that the software development and operations have followed the appropriate QA procedures.

3. GENERAL INFORMATION

3.1 SYSTEM QUALITY AND SAFETY SOFTWARE

Maintaining the integrity, safety, and security of all DOE assets and resources is paramount for DOE's mission. Since software is an integral part of DOE's resources, the integrity, safety, and security attributes of its software resources are critical to DOE's mission. All three attributes are interdependent since compromising the security access could obviously present a potential safety hazard. If the integrity of either the data or application itself has been compromised either

accidentally or maliciously, safety could be compromised. Therefore when safety software is being addressed, the integrity and security issues should likewise be addressed.

Other system level issues impacting safety software are the availability of trained and knowledgeable personnel to develop, maintain, and use the software; human factor issues such as understandability of the displays or ambient lighting conditions; and potential electromagnetic interference/radio frequency interference, which should be analyzed. Fault tolerance and common cause failure issues, performance requirements, and proper identification and analysis of functional requirements that have safety, security or integrity implications need to be propagated to the safety software.

From the foregoing, it can be seen that there are several interdependencies and tradeoffs that should be addressed when integrating software into safety systems. The necessity for robust software quality engineering processes is obvious when safety software applications are required. However, just ensuring that a "good" software engineering process or that V&V activities exist is not sufficient alone to produce safe and reliable software.[7] The life-cycle process should focus upon the safety issues in addition to the basic software quality engineering principles. Both of these concepts are detailed in this Guide.

3.2 RISK AND SAFETY SOFTWARE

Software rarely functions as an independent entity. Software is typically a component of a system much in the same way hardware, data, and procedures are system components. Therefore, to understand the risk associated with the use of software, the software function should be considered a part of an overall system function.

The consequences of software faults need to be addressed in terms of the contribution of a software failure to an overall system failure. Issues such as security, training of operational personnel, electromagnetic interference, human-factors, or system reliability *have the potential to be safety issues.* For example, if the security of the system can be compromised, then the safety software can also be compromised. Controlling access to the system is key to maintaining the integrity of the safety software. Likewise, if human factor issues such as ambient lighting conditions and ease of use for understandability are important, the risks need to be addressed either via design or training. For PLCs or network safety software applications, electromagnetic interference could offer potential risks for operation of the safety software system.

Once the software's function within the overall system's function is known, the appropriate software life-cycle and system life-cycle practices can be identified to minimize the risk of software failure on the system. Rigor can then be applied commensurate with the risk. Managing the risk appropriately is the key to managing a safety software system. Unless risks and trade-offs of either doing or not doing an activity are evaluated, there is not a true understanding of the issues involved regarding the safety software system. Obviously, time and resource constraints should be balanced with the probability of occurrence and the potential consequences versus an occurrence of the worst case scenario. If the safety systems staff zealously and

[7] Leveson, Nancy, *Safeware: System Safety and Computers,* Addison Wesley, 1995.

religiously inappropriately invokes the strictest rigor for a Level B application, then the application has the potential never to get fielded properly. On the other hand, if the process activities are only minimally or inappropriately performed for a Level A software safety application, then very adverse consequences could potentially occur for which no mitigation strategy exists. Appropriate project management is a risk management strategy and especially so for safety software applications.

3.3 SPECIAL-PURPOSE SOFTWARE APPLICATIONS

Several categories of software have a unique purpose in safety-related functions required to support DOE nuclear facility operations. This section contains an overview of the special-purpose software and the additional considerations that should be addressed by SQA programs, processes, and procedures.

3.3.1 Toolbox and Toolbox-Equivalent Software Applications

Toolbox codes represent a small number of standard computer models or codes supporting DOE safety analysis. These codes have widespread use and are of appropriate qualification for use within DOE. The toolbox codes are acknowledged as part of DOE's Safety Software Central Registry. These codes are verified and validated and constitute a "safe harbor" methodology. That is to say, the analysts using these codes do not need to present additional defense as to their qualification provided that the analysts are sufficiently qualified to use the codes and the input parameters are valid. These codes may also include commercial or proprietary design codes where DOE considers additional SQA controls are appropriate for repetitive use in safety applications and there is a benefit to maintain centralized control of the codes. The following six widely applied safety analysis computer codes have been designated as "toolbox" codes.

- ALOHA (chemical dispersion analysis)
- CFAST (fire analysis)
- EPIcode (chemical dispersion analysis)
- GENII (radiological dispersion analysis)
- MACCS2 (radiological dispersion analysis)
- MELCOR (leak path factor analysis)

The current designated "toolbox" codes and any software recognized in the future as meeting the "toolbox" equivalency criteria are no different from other custom developed safety software as defined in Section 2.1. Consequently, software of this category should be developed or acquired, maintained, and controlled applying sound software practices as described in Section 5 of this Guide.

In the future, new versions of software may be added to the Central Registry while the older versions are removed. Over time, some of the software may be retired and recommended not to be used in DOE safety analysis. Still other software may be added through the formal toolbox-equivalent process, having been recognized as meeting the equivalency criteria. Thus, the Central Registry collection of safety software applications will be expected to evolve as

software life-cycle phases, usage, and application requirements change. Appendix B addresses the process for adding new software applications and versions to, and removal of retired software from, the Central Registry.

Additional information on the detailed toolbox SQA procedures, criteria, and evaluation plan; the evaluation of the software relative to current SQA criteria (i.e., assessment of the margin of the deficiencies or "gap" analysis); user guidance documentation; description of the toolbox-equivalent process; and code-specific information may be found in the Central Registry portion of the DOE SQA Knowledge Portal (http://www.eh.doe.gov/sqa/central_registry.htm).

3.3.2 Existing Safety Software Applications

Existing software that has not been previously approved under a QA program consistent with DOE O 414.1C and has been identified as safety software should be evaluated using the graded approach framework that is described in Section 5. This software is often referred to as legacy software. In many cases this category of software originally met DOE or industry requirements, but SQA for existing software was not updated as the SQA standards were revised.

Existing safety software should be identified and controlled prior to evaluation using the graded approach framework in this Guide. The evaluation performed should be adequate to address the correct operation of the safety software in the environment it is being used. This evaluation should include (1) identification of the capabilities and limitations for intended use, (2) any test plans and test cases required demonstrating those capabilities, and (3) instructions for use within the limitations.[8] One example of this evaluation is *a posteriori* review[9] as described in American Nuclear Society (ANS) standard, ANS 10.4. Future modifications to existing safety software should meet all safety software work activities in DOE O 414.1C associated with the changes to the safety software.

3.4 CONTINUOUS IMPROVEMENT, MEASUREMENT, AND METRICS

Lord Kelvin stated "If you can not measure it, you can not improve it."[10] This truism especially applies to safety software systems. Metrics used throughout the life-cycle should bolster the confidence that the software applications will achieve their mission in a safe and reliable manner. If design, testing, or software reliability measures are unknown, then there is no assurance that the safety software has sufficient robustness to minimize the risks.

DOE O 414.1C criterion 3, *Quality Improvement*, specifies that processes should be established and implemented to detect and prevent problems. Measurements and the metrics developed from these measures can be indicators for potential future problems, and thus, steps can be initiated to prevent the occurrence. For long term avoidance of problems, continuous improvement methods should be implemented to determine the root causes and eliminate the events that could lead to a

[8] ASME NQA-1-2000, op. cit., Part II, Subpart 2.7, Section 302, p. 105.

[9] American National Standards Institute (ANSI)/American Nuclear Society (ANS) 10.4-1987 (R1998), *Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry*, ANS, 1998, Section 11, pp. 29–32.

[10] Lord Kelvin (Sir William Thomson, 1824–1907)

reoccurrence. Metrics further provide qualitative or quantitative indicators of the improvements or lack thereof when a process or work activity has been modified. Metrics are the evidence that an improvement has occurred. Both Institute of Electrical and Electronic Engineers (IEEE) Standards 982.1 and 982.2 provide recommendations for what metrics to use and when in the software life-cycle phase applying the metric is most appropriate.

3.5 USE OF NATIONAL/INTERNATIONAL STANDARDS

Title 10 CFR 830 Subpart A and DOE O 414.1C require the use of standards to develop and implement a QAP. National/international standards facilitate a common software quality engineering approach to developing or documenting software based upon a consensus of experts in the particular topic areas. Many national and international standards bodies have developed software standards to ensure that the recognized needs of their industry and users are satisfactorily met.

In the United States, ASME is the nationally accredited body for the development of nuclear facility QA standards. DOE O 414.1C cites ASME NQA-1-2000 or other national or international consensus standards that provide an equivalent level of quality assurance requirements as ASME NQA-1-2000 as the appropriate standard for QAPs applied to nuclear-related activities (e.g., safety software). The ten QA criteria in 10 CFR 830 Subpart A and DOE O 414.1C are mapped to ASME NQA-1-2000 in Appendix C. DOE O 414.1C also requires that additional standards be used to address specific work activities conducted under the QAP, such as safety software.

In the case of ASME NQA-1-2000, Part I,[11] the requirements generally apply to safety software work activities. For example, Requirements 3, 4, 7, 11, 16, and 17 for Design Control, Procurement Document Control, Control of Purchased Items and Services, Test Control, Corrective Action, and Quality Assurance Records (respectively) can have specific safety software applicability. In addition, ASME NQA-1-2000, Part II, Subpart 2.7, and Part IV, Subpart 4.1, specifically address “quality assurance requirements for computer software for nuclear facility applications” and “guide on quality assurance requirements for software” (respectively). As stated in the introduction of Part II, Subpart 2.7, this subpart “provides requirements for the acquisition, development, operation, maintenance, and retirement of software.” Table 3 provides a cross reference of ASME NQA-1-2000 with the ten SQA work activities in DOE O 414.1C. Although ASME NQA-1-2000 provides excellent process guidance for a software quality engineering process for managing a software development, maintenance, or procurement or otherwise acquiring software, the detailed guidance for safety software is not provided within this standard.

Appendix D of this Guide includes references to other standards useful in achieving compliance with 10 CFR 830 Subpart A and DOE O 414.1C for safety software work activities. It should be noted that the use of the standards discussed can aid in the development of a robust safety software quality engineering process and a resulting software product that is adequate for all the safety software applications. Use of consensus standards can promote a robust safety software quality engineering process and a resulting software product that is adequate for safety software applications.

[11] ASME NQA-1-2000, op.cit., Part I.

4. RECOMMENDED PROCESS

Recognizing that there are five safety software applications types within DOE listed in Section 2.1, the safety software analyst needs a defined process to enable a determination of what needs to be accomplished for each of the respective software safety applications. In addition, the safety software analyst needs a process to support the integration of software safety into the system safety process to improve system and software design, development and test efforts. Lastly, the process to manage each of the five application types should support the planning and coordination of the software safety tasks based on established priorities. Appendix E of this Guide presents the details of a risk-based graded approach for the analysis and safety software management process for (1) custom developed, (2) configurable, (3) acquired, (4) utility calculations, and (5) commercial design and analysis tools.

5. GUIDANCE

5.1 SOFTWARE SAFETY DESIGN METHODS

Safety should be designed into a system, just as quality should be built into the system. Safe design of a system, in which safety software is a subcomponent, uses two primary approaches: (1) applying good engineering practices based upon industry proven methods and (2) guiding design through the results of hazard analysis. Identifying and assessing the hazards is not enough to make a system safe. The information from hazard analysis needs to be factored in the design.[12]

Applying industry accepted software engineering and software quality engineering practices is generally the first approach to developing high quality software systems. These practices can be applied to safety software to improve the quality and add a level of assurance that the software performs its safety functions as intended. DOE O 414.1C requires SQA work activities, referred to as work activities, to be performed for safety software. Many national and international consensus standards, such as ASME NQA-1-2000, ANS 10.4, and the IEEE software engineering series provide detailed guidance for performing the work activities. Section 3.5 of this Guide describes some of these standards.

Software process capability models such as the Software Engineering Institute's legacy Software Capability Maturity Model (SW-CMM) and the more integrated model, Capability Maturity Model Integration (CMMI), are proven tools to assist in the selection of practices to perform for achieving a level of assurance that the processes performed will produce the desired level of quality for safety software. The CMMI has two approaches: staged and continuous. For organizations introducing a software process improvement program, these models should be considered.

[12] Leveson, op. cit., p. 398.

Table 3. ASME NQA-1-2000 Cross Reference to DOE Software Quality Assurance (SQA) Work Activities

SQA Work Activities / ASME NQA-1	Software (sw) project management & quality planning	Sw risk management	Sw configuration management	Procurement & supplier management	Sw requirements identification & management	Sw design & implementation	Sw safety	Verification & validation	Prblm reporting & corrective action	Training of … safety sw
Organization	Req. 1, 1A-1, *200*	2A-2, *301*								
Quality Assurance Program	SP 2.7, *400* 2A-2, *300* 1A-1, *200*	SP 2.7, *400* 2A-2, *301, 502*			Req. 1 Req. 2, *100* SP 2.7, *400* SP 4.1, *400*		Req. 2 SP 2.7, *402*	1A-1, *303*		Req. 2, 2A-2, *600*
Design Control		SP 2.7, *400* SP 4.1, *101, 200, 404, 406*	Req. 3, *802* SP 2.7, *203* SP 4.1, *203*	SP 2.7, *300* SP 4.1, *300*	Req. 3, *800* SP 2.7, *400* SP 4.1, *400*	Req. 3, *800* SP 2.7, *400* SP 4.1, *400*	Req. 3, *800* SP 2.7, *402* SP 4.1, *100*	Req. 3, *800* *801.4*, *801.5* Req. 11, *400* SP 2.7, *402.1, 404* SP 4.1, *402.1, 404*	Req. 15 Req. 16 SP 2.7, *204* SP 4.1, *204*	
Procurement Document Control				Req. 4						
Instructions, Procedures, and Drawings	Req. 5				Req. 5					
Document Control			Req. 6 SP 2.7, *201* SP 4.1, *201*		Req. 6 SP 2.7, *201* SP 4.1, *201*					

SQA Work Activities / ASME NQA-1	Software (sw) project management & quality planning	Sw risk management	Sw configuration management	Procurement & supplier management	Sw requirements identification & management	Sw design & implementation	Sw safety	Verification & validation	Prblm reporting & corrective action	Training of … safety sw
Control of Purchased Items and Services				Req. 7, SP 2.7, *300* SP 4.1, *300*						
Identification and Control of Items				Req. 3, *802* SP 2.7, *203* SP 4.1, *203*						
Control of Special Processes										
Inspection								Req. 3, *801.4, 801.5* Req. 11, *400* SP 2.7, *402.1, 404* SP 4.1, *402.1, 404*		
Test Control								Req. 3, *801.4, 801.5* Req. 11, *400* SP 2.7, *402.1, 404* SP 4.1, *402.1, 404*		

SQA Work Activities / ASME NQA-1	Software (sw) project management & quality planning	Sw risk management	Sw configuration management	Procurement & supplier management	Sw requirements identification & management	Sw design & implementation	Sw safety	Verification & validation	Prblm reporting & corrective action	Training of … safety sw
Control of Measuring and Test Equipment								SP 4.1, *101.3*		
Handling, Storage, and Shipping										
Inspection, Test and Operating Status										
Control of Nonconforming Items									Req. 15 SP 2.7, *204* SP 4.1, *204*	
Corrective Action									Req. 16 SP 2.7, *204* SP 4.1, *204*	
Quality Assurance Records	SP 2.7, *201* SP 4.1, *201*									
Audits	Req. 18									

For safety systems, hazards and accident analyses are performed at the system level and then for any subcomponent of the system that potentially could have an adverse effect on safety. Since software is a subcomponent of the system, hazard analysis specific to the safety software is performed. Hazard analysis is best performed periodically throughout the life-cycle of the safety software development and operations to reassess the hazards and safety of the system and its software. The information from these hazard analyses is used to make design decisions related to the safety software and its associated safety system.

5.2 SOFTWARE WORK ACTIVITIES

Software should be controlled in a traceable, planned, and orderly manner. The safety software quality work activities defined in this section provide the basis for planning, implementing, maintaining, and operating safety software. The work activities for safety software include tasks such as software project planning, SCM, and risk analysis that cross all phases in the life-cycle. Additionally, the work activities include tasks that are specific to a life-cycle phase. These work activities cover tasks during the development, maintenance, and operations of safety software.

The work activities should be implemented based upon the graded level of the safety software and the applicable software type. Table 4 provides a summary of the mapping between software type, the grading levels, and the ten SQA work activities. Not all work activities will be applicable for a particular instance of safety software. This Guide indicates where these work activities may be omitted. However, the best judgment of the software quality engineering and safety system staffs should take precedence over any optional work activities presented in this Guide.

5.2.1 Software Project Management and Quality Planning

As with any system, project management and quality planning are key elements to establishing the foundation to ensure a quality product that meets project goals. For software, project management starts with the system level project management and quality planning. Software specific tasks should be identified and either included within the overall system planning or in separate software planning documents.

These tasks may be documented in a software project management plan (SPMP), an SQA plan (SQAP), a software development plan (SDP), or similar documents. They also may be embedded in the overall system level planning documents. Typically the SPMP, SQAP, and/or SDP are the controlling documents that define and guide the processes necessary to satisfy project requirements, including the software quality requirements. These plans are initiated early in the project life-cycle and are maintained throughout the life of the project.[13]

The software project management and quality planning should include identifying all tasks associated with the software development and procurement, including procurement of services,

[13] Capability Maturity Model Integration (CMMI) Product Team, Capability Maturity Model Integration, Version 1.1, CMMI for Software Engineering (CMMI-SW, V1.1), Staged Representation, CMU/SEI-2002-TR-029, ESC-TR-2002-029, Carnegie Mellon University, Software Engineering Institute, 2002.

Table 4. Mapping Safety Software Types and Grading Levels to Software Quality Assurance (SQA) Work Activities

SQA Work Activity	Level A					Level B							Level C		
	Custom Developed	Configurable	Acquired	Utility Calcs	Commercial D & A	Custom Developed	Configurable	Acquired	Utility Calcs	Commercial D & A	Custom Developed	Configurable	Acquired	Utility Calcs	Commercial D & A
Software (Sw) project management & quality planning	Full	Full	Grade	Grade	n/a	Full	Full	Grade	Grade	n/a	Grade	Grade	Grade	Grade	n/a
Sw risk management	Full	Full	Full	Full	n/a	Grade	Grade	Grade	Grade	n/a	Grade	Grade	Grade	Grade	n/a
Sw configuration management	Full	Grade	Grade	Grade	Grade	Full	Grade	Grade	Grade	Grade	Grade	Grade	Grade	Grade	Grade
Procurement & supplier management	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full
Sw requirements identification & management	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full
Sw design & implementation	Full	Grade	n/a	Grade	n/a	Full	Grade	n/a	Grade	n/a	Full	Grade	n/a	Grade	n/a
Sw safety	Full	Full	Full	n/a	n/a	Grade	Grade	Grade	n/a	n/a	Grade	Grade	Grade	n/a	n/a
Verification & Validation	Full	Full	Full	Grade	n/a	Grade	Grade	Grade	Grade	n/a	Grade	Grade	Grade	Grade	n/a
Problem reporting & corrective action	Full	Full	Full	Grade	Full	Full	Full	Full	Grade	Full	Full	Grade	Grade	Grade	Grade
Training of … safety Sw	Full	Full	Full	Full	n/a	Grade	Grade	Grade	Grade	n/a	Grade	Grade	Grade	Grade	n/a

estimate of the duration of the tasks, resources allocated to the task, and any dependencies. The planning should include a description of the tasks and any relevant information. In addition to NQA-1-2000, several consensus standards[14,15] provide details of planning documents that are good resources to assist in the identification and description of the software development and procurement tasks.

Software quality and software development planning identifies and guides the software phases and any grading of the SQA and software development activities performed during software development or maintenance. The software quality and software engineering activities and rigor of implementation will be dependent on the identified grading level of safety software and the ability of DOE or its contractors to build quality in and assess the quality of the safety software. Because SQAP and SDP are overall basic quality and software engineering plans, some quality activities, such as SCM, risk management, problem reporting and corrective actions, and V&V, including software reviews and testing, may be further detailed in separate plans. These plans and the activities identified in these plans will be discussed later in this Guide.

Software project management and quality planning fully apply to custom developed and configurable software types for both Level A and Level B safety software. For Level A and Level B acquired and utility calculation and all Level C software applications, software project management and quality planning tasks can be graded. This grading should include the identification and tracking of all significant software tasks. Where instances of the software life-cycle may include little or no software development activities, the software project and quality planning will most likely be part of the overall system level project or facility planning. This work activity does not apply to commercial design and analysis software because the project management and quality planning activities associated with commercial design and analysis software are performed by the service supplier. DOE controls the SQA activities of that software through procurement agreements and specifications.

5.2.2 Software Risk Management

Software risk management provides a disciplined environment for proactive decision making to continuously assess what can go wrong, determine what risks are important to address, and implement actions to address those risks.[16] Because risk management is such a fundamental tool for project management, it is an integral part of software project management. Although sometimes associated with safety analysis of potential failures, software risk management focuses on the risks to the successful completion of the software project. The risks addressed by this work activity are project management risks associated with the successful completion of a software application whereas Section 5.2.7 addresses the risks associated with the potential failure modes of the software.

---

[14] Institute of Electrical and Electronic Engineers (IEEE), Std 1058-1998, *IEEE Standard for Software Project Management Plans*, IEEE, 1998.

[15] IEEE Std 730-2002, *IEEE Standard for Software Quality Assurance Plans*, IEEE, 2002.

[16] SQAS21.01.00-1999 (Reference Document), *Software Risk Management: A Practical Guide*, Department of Energy Quality Managers Software Quality Assurance Subcommittee, dated 2-2000.

Risk assessment and risk control are two fundamental activities required for project success. Risk assessment addresses identification of the potential risks, analysis of those risks, and prioritizing the risks to ensure that the necessary resources will be available to mitigate the risks. Risk control addresses risk tracking and resolution of the risks. Identification, tracking, and management of the risks throughout all phases of the project's life-cycle should include special emphasis on tracking the risks associated with costs, resources, schedules, and technical aspects of the project. Several risk identification techniques are described and detailed in standards and literature.[17,18]

Risk resolution includes risk avoidance, mitigation, or transference. Even the small risks during one phase of the safety software application's life have the potential to increase in some other phase of the application's life with very adverse consequences**.** In addition, mitigation actions for some risks could create new (secondary) risks**.**

Examples of potential software risks for the safety software application might include—

- incomplete or volatile software requirements;
- specification of incorrect or overly simplified algorithms or algorithms that will be very difficult to address within safety software;
- hardware constraints that limit the design;
- potential performance issues with the design;
- a design that is based upon unrealistic or optimistic assumptions;
- design changes during coding;
- incomplete and undefined interfaces;
- using unproven computer and software technologies such as programming languages not intended for the target application;
- use of a programming language with only minimal experience using the language;
- new versions of the operating system;
- unproven testing tools and test methods;
- insufficient time for development, coding, and/or testing;
- undefined or inadequate test acceptance criteria; and
- potential quality concerns with subcontractors or suppliers.

The risks associated with the safety software applications need to be understood and documented. The above bulleted list identifies a few potential risks associated with safety software applications. Each risk should be evaluated against its risk thresholds. Different

[17] Christensen, Mark J., and Richard H. Thayer, *The Project Manager's Guide to Software Engineering's Best Practices***,** Institute of Electrical and Electronics Engineers Computer Society Press, 2001, pp. 417–447.

[18] Society of Automotive Engineers (SAE) JA1003, *Software Reliability Program Implementation Guide*, SAE 2004, Appendix C4.6.

techniques may be used to evaluate the risks. Examples of these techniques include decision trees, scenario planning, game theory, probabilistic analysis, and linear programming. Various treatment alternatives to addressing risk should be considered to avoid, reduce, or transfer risks.

Flexibility may need to be applied regarding risk management based upon the risk categorization of the safety software application. For a Level A safety software, all apparent risks known at the time, whether large or small, should be identified, analyzed for impact and probability of occurrence, prioritized, resolved to an acceptable level of risk, and tracked through the life of the safety software. For Level B or Level C software applications, the granularity for the risks to be identified, analyzed, prioritized, resolved to an acceptable level of risk, and tracked should be determined by the safety system staff and can be graded. The safety system staff should focus on the adverse events that would dominate the risk and assess these in a qualitative manner. The safety system staff also has the responsibility to determine a graded approach for resolving the risks and the process for tracking the risks.

This work activity does not apply to commercial design and analysis safety software because this software is used in conjunction with the design and analysis services provided to DOE from a commercial contractor. The risk management work activity associated with that software is performed by the service supplier. DOE controls the SQA activities of that software through procurement agreements and specifications of design or analysis requirements.

Further guidance beyond that in NQA-1-2000 regarding risk management is provided by IEEE Standard 16085-2004.[19] This standard provides guidance regarding the risk management of acquired, developed, operational, or maintained systems to support the existing organizational risk management processes. SQAS21.01.00-1999, *Software Risk Management: A Practical Guide,* also discusses a risk taxonomy, risk transference, and risk avoidance that may be of interest to the safety software analyst.

5.2.3 Software Configuration Management

SCM activities identify all functions and tasks required to manage the configuration of the software system, including software engineering items, establishing the configuration baselines to be controlled, and software configuration change control process.[20] The following four areas of SCM[21] should each be addressed when performing configuration management:
(1) configuration identification, (2) configuration control, (3) configuration status accounting, and (4) configuration audits and reviews. This Guide extends ASME NQA-1-2000 software configuration management[22] tasks by including configuration audits and reviews.[23]

---

[19] International Organization for Standardization (ISO)/Institute of Electrical and Electronics Engineers (IEEE) Std 16085, *IEEE Standard for Software Engineering: Software Life Cycle Processes—Risk Management,* IEEE, 2004.

[20] ASME NQA-1-2000, op. cit., Part II, Subpart 2.7, Section 203, p. 105.

[21] IEEE Std 828-1998, *IEEE Standard for Software Configuration Management Plans,* IEEE, 1998, Section 4.3.

[22] ASME NQA-1-2000, op. cit., Part I, Section 802, p. 16.

[23] IEEE Std 7-4.3.2-2003, *IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations,* IEEE, 2003, Section 5.3.5.

The methods used to control, uniquely identify, describe, and document the configuration of each version or update of software and its related documentation should be documented. This documentation may be included in a SCM plan or its equivalent. Such documentation should include criteria for configuration identification, change control, configuration status accounting, and configuration reviews and audits.

During operations, authorized users lists can be implemented to ensure that the software use is limited to those persons trained and authorized to use the software. Authorized users lists are access control specifications that are addressed in Section 5.2.5, Software Requirements Identification and Management.

A baseline labeling system should be implemented that uniquely identifies each configuration item, identifies changes to configuration items by revision, and provides the ability to uniquely identify each configuration. This baseline labeling system is used throughout the life of the software development and operation.

Proposed changes to the software should be documented, evaluated, and approved for release. Only approved changes should be made to the software that has been baselined. Software verification activities should be performed for the change to ensure the change was implemented correctly. This verification should also include any changes to the software documentation.

Audits or reviews should be conducted to verify that the software product is consistent with the configuration item descriptions in the requirements and that the software, including all documentation, being delivered is complete. Physical configuration audits and functional configuration audits are examples of audits or reviews that should be performed.[24] SCM work activities should be applied beginning at the point of DOE's or its contractor's control of the software.

For custom developed safety software graded at Level A or Lever B, all four areas of SCM noted above apply. For all other types of safety software graded as Level A or Level B and all Level C safety software, this work activity may be graded by the optional performance of configuration audits and reviews.

5.2.4 Procurement and Supplier Management

Most software projects will have procurement activities that require interactions with suppliers regardless of whether the software is Level A, B, or C. Procurement activities may be as basic as the purchase of compilers or other development tools for custom developed software or as complicated as procuring a complete safety system software control system. Thus, there are a variety of approaches for software procurement and supplier management based upon—

- the level of control DOE or its contractors have on the quality of the software or software service being procured and
- the complexity of the software.

[24] Institute of Electrical and Electronics Engineers (IEEE) 1042-1987, *IEEE Guide to Software Configuration Management,* IEEE, 1987, Section 3.3.4.

Procurement documentation should include the technical[25] and quality[26] requirements for the safety software. Some of the specifications that should be included are—

- specifications for the software features, including requirements for safety, security, functions, and performance;
- process steps used in developing and validating the software, including any documentation to be delivered;
- requirements for supplier notification of defects, new releases, or other issues[27] that impact the operation; and
- mechanisms for the users of the software to report defects and request assistance in operating the software.

These requirements should be assessed for completeness and to ensure the quality of the software being purchased. There are four major approaches for this assessment:

- performing an assessment of the supplier,
- requiring the supplier to provide a self-declaration that the safety software meets the intended quality,
- accepting the safety software based upon key characteristics (e.g., large user base), and
- verifying the supplier has obtained a certification or accreditation of the software product quality or software quality program from a third party (e.g., the International Organization for Standardization, Underwriters Laboratories, and Software Engineering Institute).

It is important to note that while Levels A, B, and C software applications are required to fully meet this work activity, the implementation detail and assessment method of the supplier can vary based on the complexity of the software and its importance to safety.

5.2.5 Software Requirements Identification and Management

Safety system requirements provide the foundation for the requirements to be implemented in the software. These system requirements should be translated into requirements specific for the software. The identified software requirements may be documented in system level requirements documents, software requirements specifications, procurement contracts, and/or other acquired software agreements. These requirements should identify functional; performance; security, including user access control; interface and safety requirements; and installation considerations and design constraints where appropriate. The requirements should be complete, correct, consistent, clear, verifiable, and feasible.[28]

---

[25] ASME NQA-1-2000, op. cit., Part I, Requirement 4, Section 202, p. 18.

[26] ASME NQA-1-2000, op. cit., Part I, Requirement 4, Section 100, p.18.

[27] ASME NQA-1-2000, op. cit., Part II, Subpart 2.7, Section 301, p. 105.

[28] Institute of Electrical and Electronics Engineers (IEEE) Std 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*, IEEE, 1998, Section 4.3.

User access control during operations is an important aspect to ensuring only authorized users can operate the system or use the software for design or analysis tasks. Controlling access is a software safety and/or security requirement that can be associated with training or qualification to operate the system. ASME NQA-1-2000 addresses access control specifications as part of the operations phase.[29]

Once the software requirements have been defined and documented, they should be managed to minimize conflicting requirements and maintain accuracy for later validation activities to ensure the correctness of the software placed into operations. Software requirements should be traceable throughout the software life-cycle.[30]

This work activity has no grading associated with its performance. Software requirements identification management and traceability applies to Level A, B, and C software applications and should fully meet this requirement. However, the detail and format of the safety software requirements may vary with the software type. Custom developed software most likely will contain a larger number of software requirements than configurable, acquired, utility calculation, or commercial design and analysis tool software, and thus, a separate more formal document may be applicable.

5.2.6 Software Design and Implementation

During software design and implementation the software is developed, documented, reviewed, and controlled. The software design elements should identify the operating system, function, interfaces, performance requirements, installation considerations, design inputs, and design constraints. The software design should be complete and sufficient to meet the software requirements.[31] The design activities and documentation should be adequate to fully describe how the software will interface with other system components[32] and how the software will function internally. Data structure requirements and layouts may be necessary to fully understand the internal operations of the software.

Custom developed software will require more formality in the documentation and review of the design than configurable or utility calculations. Simple process flows, relationships between data elements, interfaces with external components, and basic database table structures may be all that are needed for configurable or utility calculations, whereas for custom developed software, complete functional and logical designs of the software components, the input and output data, and pseudo code may be required to fully understand the safety software design. The software design description may be combined with the documentation of the software requirements or software source code.[33]

During implementation, static analysis, clean room inspections, and reviews are common techniques to ensure the implementation remains consistent with the design and does not add

[29] ASME NQA-1-2000, op. cit., Part II, Subpart 2.7, Section 405, p. 106.
[30] ASME NQA-1-2000, op. cit., Part II, Subpart 2.7, Section 401, p. 106.
[31] ASME NQA-1-2000, op. cit., Part I Introduction, and Section 801.2, p. 16.
[32] ASME NQA-1-2000, op. cit., Part II, Subpart 2.7, Section 402, p. 106.
[33] ASME NQA-1-2000, op. cit., Part I Introduction and Section 801.2, p. 16.

complexity or functions which could decrease the safe operation of the software. Many tools exist to evaluate the complexity and other attributes of the source code design structure. Walkthroughs and more formal inspections, such as Fagan inspections, can be used to identify defects in source code, as well as design descriptions and other software development process outputs.

The software developer should perform unit testing prior to system level V&V techniques, including acceptance testing. Developer testing can be very structured and formal, using automated tools or less formal methods. In addition to unit testing, functional, structural, timing (performance testing), stress, security, and human-factors testing are useful testing methods. These methods can be applied using a graded or tailored approach to ensure the known risks are mitigated appropriately. Other techniques[34,35] such as error seeding; equivalence class testing; branch and path testing; statistical-based, boundary value testing; and code coverage analysis may all be beneficial testing techniques to ensure robust and reliable software.

The software design and implementation work activity for Levels A, B, and C custom developed software applications should fully meet this requirement. For this software type, the design, including interfaces and data structures, should be completely documented; reviews of the design and code should be performed. Additionally, formal developer testing that includes functional, structural, timing, stress, security, and human-factors testing should be planned, performed and the results documented. It is recommended that the complexity of the custom developed safety software be evaluated and analysis performed to reduce the complexity of the source code modules.

Configurable and utility calculation for Levels A, B, and C software applications may be graded for this work activity. This grading should include fully performing the design work activities as with custom developed software. However, less formal design and code reviews, such as simple desk checks by another individual other than the developer, may be performed. Developer testing should be performed and documented that includes safety functions, security, and performance testing. This work activity does not apply to acquired or commercial design and analysis safety software types since the design and implementation activities associated with commercial design and analysis software are performed by the service supplier. DOE controls the SQA activities of that software through procurement agreements and specifications.

5.2.7 Software Safety

The development of software applications requires identification of hazards (i.e., abnormal conditions and events) that have the potential for defeating a safety function and the implementation of design strategies to eliminate or mitigate those hazards. Hence, it is recommended that the software safety process address the mitigation strategy for the components that have potential safety consequences if a fault occurs, whereas the software design and implementation process addresses the architecture of the safety software application.

---

[34] Pressman, Roger S., *Software Engineering: A Practitioner's Approach,* McGraw Hill, 1992, pp. 595–629.

[35] Sparkman, Debra, *Techniques, Processes, and Measures for Software Safety and Reliability,* Lawrence Livermore National Laboratory, UCRL-ID 108725, 1992.

Software is only one component of the overall safety system. It may be embedded in an I&C system, it may be a custom control system for hardware components, or it may be standalone software used in safety management or support decisions. In any of these or other applications of software important to safety, analysis of the software application occurs first at the system level. The analysis should then be performed at the software component level to ensure adequate safeguards are provided to eliminate or mitigate the potential occurrence of a software defect that could cause a system failure.

Methods to mitigate the consequences of software failures should be an integral part of the software design.[36] Specific software analysis and design methods for ensuring that safety functions are well thought out and addressed properly should be performed throughout the software development and operations life-cycles. These methods include dynamic and static analyses. The techniques and methods described in this section are only a selection of those available. Several resources are available to assist in the selection and use of these methods. A few are listed in the reference section of this Guide.

During the initial concept and requirement analysis phases for the software, potential failures need to be identified and evaluated for their consequences of failure and probability of occurrence. Some potential problems are (1) complex or faulty algorithm, (2) lack of proper handling of incorrect data or error conditions, (3) buffer overflow, and (4) incorrect sequence of operations due to either logic or timing faults.

There are several hazard analysis techniques that may be used for this purpose. Many of these techniques are performed as preliminary analyses and later updated as more information is known about the requirements and design structure. These techniques include failure modes and effects analysis, fault-tree modeling, event-tree modeling, cause-consequence diagrams, hazard and operability analysis, and interface analysis. Techniques such as these should be applied and appropriately documented to understand and assess the impact of software failures on the system.

The design of the software is critical to ensuring safe operation of the system. The software design should consider principles of simplicity, decoupling, and isolation to eliminate the hazards.[37] Complexity of the software design, including the logic and number of data inputs, has proven to increase the defect density in software components. The safety features should be separate from nonsafety modules, minimizing the impact of failure of one module on another.[38] The interfaces between the modules need to be defined and tested thoroughly. Separation of the safety features also allows for more rigorous software development and verification practices to be applied to the safety components while providing the appropriate and cost effective level of SQA applied to the nonsafety components. Software engineering safety design practices should include process flow analysis, data flow analysis, path analysis, interface analysis, and interrupt analysis during the design phase.

[36] ASME NQA-1-2000, op. cit., Part II, Subpart 2.7, Section 402, p. 106.
[37] Leveson, op. cit., pp. 400–412.
[38] IEEE Std. 7-4.3.2-2003, op. cit., Section 5.6, p. 13.

When hazards related to software functions cannot be eliminated, the hazard should be reduced and/or monitored. Additionally, software can experience partial failures that can degrade the capabilities of the overall system that may not be immediately detectable by the system. In these instances, other design techniques, such as building fault detection and self-diagnostics into the software, should be implemented. Using external monitors (safety bag) for the software safety functions, n-version programming, and Petri nets are examples of techniques[39,40] that can ensure the software design adequately addresses safety issues and minimizes failure modes by adding fault tolerant concepts. Self-diagnostics detect and report software faults and failures in a timely manner and allow actions to be taken to avoid an impact on the system operating safety. Some of these techniques include memory functionality and integrity tests, such as checksums and watch dog timers for software processes, including operating system processes.[41] Additionally, software control functions can be performed incrementally rather than in a single step, reducing the potential that a single failure of a software component would cause an unsafe state.

The software safety work activity for Level A custom developed, configurable, and acquired safety software should fully meet this requirement. For this software type the safety analysis for the software components should be performed. This analysis may be part of the overall safety system analysis if detailed software failures are included. For Level A custom developed safety software, the design concepts that include simplicity of modules that perform safety functions and isolation of those modules should be part of the design considerations. Where the design of the software modules still presents an unacceptable risk to failure of the safety system, fault tolerant and self-diagnostics designs should be implemented.

Custom developed, configurable, and acquired Level B or Level C software applications may be graded. This grading may include fully performing the safety analysis activities for the software components to ensure the safety aspects are being addressed. The design concepts of simplicity and isolation and fault tolerance and self-diagnostics may not apply to Level B or Level C software applications and, thus, can optionally be applied.

This work activity does not apply to utility calculation or commercial design and analysis safety software types. Utility calculations are typically simple calculations where techniques and methods described above could add undue burden to the development of these applications and not increase the assurance that any software failure would not impact safety. However, if the safety analysis determines that complexity of the utility calculation warrants the use of these techniques, they should be applied. For commercial design and analysis software, the software safety activities are performed by the service supplier. DOE controls the SQA activities of that software through procurement agreements and specifications.

5.2.8 Verification and Validation

V&V is the largest area within the SQA work activities. Verification is performed throughout the life-cycle of the safety software. Validation activities are performed at the end of the software development or acquisition processes to ensure the software meets the intended requirements.

[39] Sparkman, op. cit.
[40] SAE JA1003, op. cit., Appendix C.
[41] IEEE Std 7-4.3.2-2003, op. cit., Section 5.5.3, p. 13.

V&V activities should be performed by competent staff other than those who developed the item being verified or validated.[42] V&V activities include reviews, inspections, assessments, observations, and testing. This Guide expands on ASME NQA-1-2000 acceptance testing activities to include more extensive V&V activities of reviews, inspections, assessments, and observations as described in other consensus standards.

Reviews and inspections of software deliverables requirement specifications, procurement documents,[43] software design, code modules, test results, training materials, user documentation, and processes that guide the software development activities should be performed. The software deliverables may be combined with other software or system documents. Traceability of the software requirements to the software design should be performed.[44] As mentioned in the development practice section, inspections can be formally implemented Fagan inspections, walkthroughs, or desk checks. Verification of the software design, using one of the above methods, should be completed prior to approval of the software for use.[45] This verification may be performed as part of the software development and implementation activity.

Supplier assessments are important aspects of V&V. Assessments are covered in Section 5.2.4, Procurement and Supplier Management, and Section 6, Assessment and Oversight.

Observations and testing can be performed during the development, factory or site acceptance testing, installation, and operation (i.e., in-use testing)[46] of the software. Observations and testing during development is discussed in Section 5.2.6, Software Design and Implementation. Software testing activities should be planned and documented. Test cases and procedures, including expected results, should be created. All test activity deliverables should be under configuration management. Test results should be documented and all test activity deliverables placed under configuration management.[47]

Acceptance testing should include functional testing, performance testing, security testing, stress testing, and load testing. Users' guides, use cases, and operational profiles are instrumental in identifying and detailing the positive test cases and procedures. Failure mode analyses can be used for defining negative test cases and procedures. Testing strategies that may be appropriate for acceptance testing include equivalence class testing, branch and path testing, statistical-based and boundary value testing.

Additionally, the system should continually be monitored to estimate its continuing reliability and safety. Periodic testing of the operational system should be performed to detect any degradation.[48] If testing is not possible, monitoring using quantitative measurements should be performed.

---

[42] ASME NQA-1-2000, op. cit., Part I, Requirement 3, Section 801.1, p. 16.
[43] ASME NQA-1-2000, op. cit., Part I, Requirement 4, Section 300, p. 18.
[44] ASME NQA-1-2000, op. cit, Part II, Subpart 2.7, Section 402.1, p. 106.
[45] ASME NQA-1-2000, op. cit., Part II, Subpart 2.7, Section 402.1, p. 106.
[46] ASME NQA-1-2000, op. cit., Part I, Requirement 11, Section 400, p. 29.
[47] ASME NQA-1-2000, op. cit., Part I, Requirement 11, Section 200, p. 29.
[48] ASME NQA-1-2000, op. cit., Part I, Requirement 11, Section 400, p. 30.

When a new version of a software product is obtained, predetermined and ad-hoc test cases and procedures should be performed to validate that the system meets the requirements and does not perform any unintended functions.[49] If the system is operational, only positive testing may be possible. In those instances, it is important to perform analysis of failure modes for the software to understand the consequences if the software or system should get into an abnormal operational state.

Modern utility calculation applications, such as spreadsheet programs, have grown dramatically in power, with a corresponding growth in risk. The addition of macro programming languages and the ability to incorporate "add-in" programs provide users with nearly the same capabilities as code developed with traditional programming tools. Utility calculation applications are installed on virtually every desktop, and user files containing algorithms and data can be easily modified by users. Section 101.1 of ASME NQA-1-2000, Subpart 4.1, provides useful guidance on V&V of utility calculations. Calculations performed using applications such as commercial spreadsheet programs may be treated in either of two ways. In the case of relatively straightforward calculations, the calculation result may be checked and verified in the same manner as a hand calculation. For more complex or extensive calculations, where checking and verification of calculation results are impractical or undesirable, the user files containing the calculation formulas, algorithms, or macros should be subject to the entire software life-cycle process. The latter approach may also be expedient for calculation applications that are reused frequently.[50]

Custom developed software will most likely have a larger number and more detailed deliverables than would utility calculations. For Level A safety software all deliverables should be reviewed using V&V methods. Additionally for Level A, traceability of the requirements to the design and from requirements to test cases should be performed. For Level B safety software, deliverables that include requirements, test plans and procedures, and test results should be reviewed using V&V methods.

For all Level A safety software except utility calculations, acceptance testing work activities should be planned and documented; acceptance test cases and procedures, including expected results should be created; test results should be documented; and all test activity deliverables should be under configuration management. Level A utility calculations and Level B and C custom developed, configurable, acquired, and utility calculations can use a graded approach by applying less formality in the documentation. Simple check lists for acceptance test cases and procedures may be used in place of more detailed test cases and procedures. Test results should be documented and all test activity deliverables placed under configuration management.

For Level A software, continual monitoring of safety software operations based upon historical failure data and results of periodic reassessment of hazards should be performed. For Level A, B, or C software, when new releases of the safety software have been developed, reviews and acceptance testing of changed documents and software should be performed.

---

[49] ASME NQA-1-2000, op. cit., Part II, Subpart 2.7, Section 404, p. 106.
[50] ASME NQA-1-2000, op. cit., Section 101.1, Subpart 4.1.

This work activity does not apply to commercial design and analysis safety software types since the V&V activities associated with commercial design and analysis software are performed by the service supplier. DOE controls the SQA activities of that software through procurement agreements and specifications.

5.2.9 Problem Reporting and Corrective Action

Coupled with the configuration management of the software system, the problem reporting and corrective action process should address the appropriate requirements of the QAP corrective action system. The reporting and corrective action system will cover (1) methods for documenting, evaluating and correcting software problems; (2) an evaluation process for determining whether a reported problem is indeed a defect or an error; and (3) the roles and responsibilities for disposition of the problem reports, including notification to the originator of the results of the evaluation.[51] If the noted problem is indeed an error, the problem reporting and corrective action system should correlate the error with the appropriate software engineering elements; identify the potential impacts and risks to past, present, and future developmental and operational activities; and support the development of mitigation strategies. After an error has been noted, all users should be apprised to ascertain any impacts upon safety basis decisions.

Procurement documents should identify the requirements for suppliers to report problems to the supplier, any required supplier response, and the method for the purchasers to report problems to the supplier.[52]

Maintaining a robust problem reporting and corrective action process is obviously vital to maintaining a reliable and vital safety software system. This problem reporting and corrective action system need not be separate from the other problem reporting and corrective action processes if the existing process adequately addresses the items in this work activity.[53]

This work activity should be fully implemented for all Level A and B software types (custom developed, acquired, configurable, and commercial design and analysis) and for Level C custom developed. This formal implementation should include documentation and tracking to closure of any problems reported for the software and authorization to perform the corrective action. A graded approach that reduces the formality of documenting problem reports and approving corrective actions taken may be applied for Level A and B utility calculation safety software and all Level C software applications except custom developed. This less formal implementation may include interoffice communications describing the problem identified and the corrective actions planned.

5.2.10 Training Personnel in the Design, Development, Use, and Evaluation of Safety Software

Training personnel in designing, developing, testing, evaluating, or using the safety software application is critical for minimizing the consequences of software failure. Although other SQA

[51] ASME NQA-1-2000, op. cit., Part II, Subpart 2.7, Section 204, p. 105.
[52] ASME NQA-1-2000, op. cit., Part II, Subpart 2.7, Section 301, p. 105.
[53] ASME NQA-1-2000, op. cit., Part IV, Subpart 4.1, Section 204, p. 229.

work activities may indicate that the software satisfies its operational objective, improper or invalid use of the software may negate the safety mitigation strategies included within the software.

Training may be necessary for the analyst, development and test teams, application users, and operations staff. The analyst and developers may need training in fault tolerant methodologies, safety design methodologies, user interface design issues, testing methodologies, or configuration management to ensure delivery of a robust software application. Meanwhile, the software application users and operations staff may need training specific to the software to ensure that proper data are entered, that proper options and menus are selected, and that the results of the software can be interpreted correctly. A trained and knowledgeable staff is essential to assess and evaluate the SQA requirements to ensure the proper levels of quality and safety exists in the software.

Training should be commensurate with the scope, complexity, and importance of the tasks and the education, experience, and proficiency of the individual. Indoctrination as described in ASME NQA-1-2000[54] meets this work activity requirement. Personnel should also participate in continuing education and training as necessary to improve their performance and proficiency and ensure that they stay up-to-date on changing technology and new requirements.[55]

Completion of training, education, and/or qualification requirements for all staff involved in the development, testing, use, and evaluation of custom developed or configurable software graded as Level A, B, or C should be documented and reviewed periodically. This may include a position description, qualification criteria, or a list of training courses along with verification of successfully meeting the knowledge requirements. Completion of training, education, and/or qualification requirements for all staff involved in the procurement, testing, use, and evaluation of acquired or utility calculation software graded as Level A should be documented and reviewed periodically. For Level B and C software applications, this work activity can be graded to include periodic evaluation by the appropriate supervising authority of the training, educational, or qualification requirements for performing assigned tasks associated with using and evaluating acquired or utility calculation software. This work activity does not apply to commercial design and analysis safety software since the training activities associated with commercial design and analysis software are performed by the service supplier. DOE controls the SQA activities of that software through procurement agreements and specifications.

6. ASSESSMENT AND OVERSIGHT

6.1 GENERAL

DOE assessment requirements in 10 CFR 830 Subpart A and DOE O 414.1C should be applied to safety software management and control issues.

[54] ASME NQA-1-2000, op. cit., Part I, Requirement 2, Section 200, p. 10.
[55] DOE-STD-1172-2003, *Safety Software Quality Assurance Functional Area Qualification Standard*, dated 12-03.

6.2 DOE AND CONTRACTOR ASSESSMENT

DOE should assess the effectiveness of its actions in resolving issues related to safety software management and controls. DOE also evaluates the adequacy and implementation effectiveness of DOE and contractor safety software management and controls. DOE G 414.1-1, *Management Assessment and Independent Assessment Guide for Use with 10 CFR, Part 830, Subpart A, and DOE O 414.1A, Quality Assurance;* DOE P 450.4, *Safety Management System Policy; and* DOE P 450.5, *Line ES&H Oversight Policy*, dated 5-31-01, contains guidance on independent and management assessment.

Contractors are expected to assess the adequacy and effectiveness of their safety software controls in accordance with DOE O 414.1C and this Guide.

A model criteria review and approach document (CRAD) is provided in Appendix F. This model contains software qualification assessment criteria for assessing the safety software used for safety analysis and design of safety SSCs and I&C systems in the defense nuclear facilities.

The organization responsible for the work will ensure that the SQA implementation process addresses the processes presented in this Guide.

6.3 DOE INDEPENDENT OVERSIGHT

The DOE Office of Independent Oversight and Performance Assurance, and the Office of Inspector General are responsible for conducting independent oversight of DOE actions related to safety software issues.

The DOE/NNSA SQA responsible person will verify that the SQA implementation process meets the intent of this Guide throughout the entire software life-cycle as described in the QAP and procedures.

APPENDIX A. ACRONYMS AND DEFINTIONS

A.1. ACRONYMS

ANS	American Nuclear Society
ANSI	American National Standards Institute
ASME	American Society of Mechanical Engineers
ASQC	American Society for Quality Control
CFR	Code of Federal Regulations
CMMI	Capability Maturity Model Integration
COTS	commercial off-the-shelf
CRAD	criteria review and approach document
DCS	distributed control system
DNFSB	Defense Nuclear Facilities Safety Board
DoD	U.S. Department of Defense
DOE G	U.S. Department of Energy Guide
DOE O	U.S. Department of Energy Order
DOE	U.S. Department of Energy
DSA	documented safety analysis
HMI	human-machine interface
I&C	Instrumentation and control
IAEA	International Atomic Energy Agency
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
M&O	management and operating
NASA	National Aeronautics and Space Administration
NNSA	National Nuclear Security Administration
PLC	programmable logic controller
QA	quality assurance
QAP	quality assurance program
QARD	quality assurance requirements document
RSICC	Radiation Safety Information Computational Center
SAE	Society of Automotive Engineers
SC	safety class
SCM	software configuration management
SDD	software design description
SDP	software development plan

SEI	Software Engineering Institute
SG	safety guide
SMS	safety management system
SPMP	software project management plan
SQA	software quality assurance
SQAP	software quality assurance plan
SRS	software requirement specification
SS	safety significant
SSC	structure, system, and component
SW-CMM	Software Capability Maturity Model
TR	technical report
TSR	technical safety requirement
USQ	unreviewed safety question
USQD	unreviewed safety question determination
V&V	verification and validation
VV&A	verification, validation, and accreditation
WIPP	Waste Isolation Pilot Project

A.2. DEFINITIONS

The following definitions are included with this Guide for convenience and clarification. DOE O 414.1C definitions shall take precedence over those included in this appendix.

Acceptance Testing. The process of exercising or evaluating a system or system component by manual or automated means to ensure that it satisfies the specified requirements and to identify differences between expected and actual results in the operating environment. Source: ASME NQA-1-2000.

Administrative Controls. The provisions relating to organization and management, procedures, record keeping, assessment, and reporting necessary to ensure safe operation of a facility. Source: 10 CFR 830.

Assessment. A review, evaluation, inspection, test, check, surveillance, or audit, to determine and document whether items, processes, systems, or services meet specified requirements and perform effectively. Source: DOE O 414.1C.

Configuration Management. The process of identifying and defining the configuration items in a system (i.e., software and hardware), controlling the release and change of these items throughout the system's life cycle, and recording and reporting the status of configuration items and change requests. Source: ASME NQA-1-2000.

Consequence. An outcome of an event, hazard, threat, or situation. Source: IEEE Std 1540-2001.

Firmware. The combination of a hardware device and computer instructions and data that reside as read-only software on that device. Notes: (1) This term is sometimes used to refer only to the hardware device or only to the computer instructions or data, but these meanings are deprecated. (2) The confusion surrounding this term has led some to suggest that it be avoided altogether. Source: IEEE Std 610.12-1990.

Functional Configuration Audit. An audit conducted to verify that the development of a configuration item has been completed satisfactorily, that the item has achieved the performance and functional characteristics specified in the functional or allocated configuration identification, and that its operational and support documents are complete and satisfactory. Source: IEEE Std-610.12-1990.

Graded Approach. The process of ensuring that the level of analyses, documentation, and actions used to comply with requirements are commensurate with—

- the relative importance to safety, safeguards, and security;
- the magnitude of any hazard involved;
- the life-cycle stage of a facility or item;
- the programmatic mission of a facility;
- the particular characteristics of a facility or item;

- the relative importance to radiological and nonradiological hazards; and
- any other relevant factors.

Source: 10 CFR 830.

Hazard Controls. Measures to eliminate, limit, or mitigate hazards to workers, the public, or the environment, including— 10 CFR 830

(1) physical, design, structural, and engineering features;
(2) safety structures, systems and components
(3) safety management programs;
(4) Technical Safety Requirements; and
(5) other controls necessary to provide adequate protection from hazards.

Source: 10 CFR 830.

Item. An all-inclusive term used in place of appurtenance, assembly, component, equipment, material, module, part, structure, product, software, subassembly, subsystem, system, unit, or support systems. Source: 10 CFR 830.

Nuclear Facility. A reactor or a nonreactor nuclear facility where an activity is conducted for or on behalf of DOE and includes any related area, structure, facility, or activity to the extent necessary to ensure proper implementation of the requirements established in CFR, part 10, section 830. Source: 10 CFR 830.

Physical Configuration Audit. An audit conducted to verify that a configuration item, as built, conforms to the technical documentation that defines it. IEEE Std-610.12-1990.

Process. A series of actions that achieves an end result. Source: 10 CFR 830.

Quality. The condition achieved when an item, service, or process meets or exceeds the user's requirements and expectations. Source: 10 CFR 830.

Quality Assurance. All those actions that provide confidence that quality is achieved. Source: 10 CFR 830.

Quality Assurance Program. The overall program or management system established to assign responsibilities and authorities, define policies and requirements, and provide for the performance and assessment of work. Source: 10 CFR 830.

Risk. The likelihood of an event, hazard, threat, or situation occurring and its undesirable consequences; a potential problem. Source: IEEE Std 1540-2001.

Safety. An all-inclusive term used synonymously with environment, safety, and health to encompass protection of the public, the workers, and the environment. Source: DOE O 414.1C.

Safety-class structures, systems, and components (SC SSCs). Structures, systems, or components, including portions of process systems, whose preventive and mitigative function is

necessary to limit radioactive hazardous material exposure to the public, as determined from the safety analyses. Source: 10 CFR 830.

Safety-significant structures, systems, and components (SS SSCs). Structures, systems, and components which are not designated as safety-class SSCs, but whose preventive or mitigative function is a major contributor to defense in depth and/or worker safety as determined from safety analyses [10 CFR 830]. As a general rule of thumb, safety-significant SSC designations based on worker safety are limited to those systems, structures, or components whose failure is estimated to result in a prompt worker fatality or serious injuries (e.g., loss of eye, loss of limb) or significant radiological or chemical exposure to workers. Source: DOE G 420.1-1

Safety and Hazard Analysis Software and Design Software. Software that is used to classify, design, or analyze nuclear facilities. This software is not part of an SSC but helps to ensure the proper accident or hazards analysis of nuclear facilities or an SSC that performs a safety function. Source: DOE O 414.1C.

Safety Management and Administrative Controls Software. Software that performs a hazard control function in support of nuclear facility or radiological safety management programs or Technical Safety Requirements or other software that performs a control function necessary to provide adequate protection from nuclear facility or radiological hazards. This software supports eliminating, limiting, or mitigating nuclear hazards to workers, the public, or the environment as addressed in 10 CFR 830, 10 CFR 835, and the DEAR ISMS clause. Source: DOE O 414.1C.

Safety Management Program. A program designed to ensure a facility is operated in a manner that adequately protects workers, the public, and the environment by covering a topic such as: quality assurance; maintenance of safety systems; personnel training; conduct of operations; inadvertent criticality protection; emergency preparedness; fire protection; waste management; or radiological protection of workers, the public, and the environment. Source: 10 CFR 830.

Safety Software. Includes safety system software, safety and hazard analysis software and design software and safety management and administrative controls software. Source: DOE O 414.1C.

Safety Structures, Systems, and Components. Both safety class structures, systems, and components and safety significant structures, systems, and components. Source: 10 CFR 830.

Safety System Software. Software for a nuclear facility[1] that performs a safety function as part of a structure, system or component and is cited in either DOE approved documented safety analysis or an approved hazard analysis per DOE P 450.4, *Safety Management System Policy*, dated 10-15-96, and the DEAR clause. Source: DOE O 414.1C.

Service. Work, such as design, construction, fabrication, decontamination, environmental remediation, waste management, laboratory sample analysis, safety software development/

[1] Per 10 CFR 830, quality assurance requirements apply to all DOE nuclear facilities including radiological facilities (see 10 CFR 830, DOE Std 1120, and the DEAR clause).

validation/testing, inspection, nondestructive examination/testing, environmental qualification, equipment qualification, training, assessment, repair, and installation or the like. Source: 10 CFR 830.

Software. Computer programs, procedures, and associated documentation and data pertaining to the operation of a computer system. Source: NQA-1-2000

Technical Safety Requirements. The limits, controls, and related actions that establish the specific parameters and requisite actions for the safe operation of a nuclear facility and include, as appropriate for the work and the hazards identified in the documents safety analysis for the facility: safety limits, operating limits, surveillance requirements, administrative and management controls, use and application provisions, and design features, as well as a bases appendix. Source: 10 CFR 830.

Verification and Validation. The process of determining whether the requirements for a system or component are complete and correct, the products of each development phase fulfill the requirements or conditions imposed by the previous phase, and the final system or component complies with specified requirements. Source: IEEE Std-610.12-1990.

Work. A defined task or activity; such as research and development; operations; environmental remediation; maintenance and repair; administration; safety software development, validation, testing, and use; inspection; safeguards and security; or data collection and analysis. Source: DOE O 414.1C.

APPENDIX B. PROCEDURE FOR ADDING OR REVISING SOFTWARE TO OR DELETING SOFTWARE FROM THE DOE SAFETY SOFTWARE CENTRAL REGISTRY

CONTENTS

PROCEDURE FOR ADDING OR REVISING SOFTWARE TO OR DELETING SOFTWARE FROM THE DOE SAFETY SOFTWARE CENTRAL REGISTRY

B.1 INTRODUCTION

B.1.1 PURPOSE

Toolbox codes represent a small number of standard computer models or codes supporting DOE safety analysis having widespread use and of appropriate qualification that are maintained, managed and distributed by DOE's Safety Software Central Registry (referred to as the Central Registry). The purpose of this appendix is to outline the procedure for adding new software to the Central Registry that is consistent with software quality assurance (SQA) requirements of DOE O 414.1C, *Quality Assurance*,[1] Criteria are referenced for demonstrating compliance with applicable SQA requirements, and are recommended for use in an evaluation process to determine suitability of candidate software for inclusion in the Central Registry. Information is also presented in brief on the procedures to (1) revise or update toolbox software and (2) remove software from the Central Registry due to retirement by the software developer.

More detailed information of the SQA requirements and criteria that are applicable to safety software as a basis for consideration to the Central Registry is found at http://www.eh.doe.gov/sqa/central_registry.htm.

B.1.2 SCOPE

The scope of this procedure includes any software application used by DOE or its contractors for a safety-related purpose that is proposed for inclusion in the Central Registry.

B.1.3 FUNCTIONS

Procedures to identify, document and submit additional software applications to the Central Registry are based on the process followed to evaluate the six initial toolbox codes.[2] Following this precedent, three principal entities described below perform the major tasks.

Software Sponsor—either the originator of the software (developer) or the primary user (site organization) who is requesting the software to be placed in the Central Registry or a combination of the two. In either case, this party is responsible for documenting SQA programs, procedures and processes associated with development of the software, maintaining and configuration controlling the software, developing new versions of the subject software, addressing user questions, and resolving technical and programmatic issues. The software sponsor is responsible for documenting the rationale for adding the subject software to the Central Registry.

---

[1] DOE O 414.1C, *Quality Assurance*, dated 6-17-05.

[2] U.S. Department of Energy, *Implementation Plan for Defense Nuclear Facilities Safety Board Recommendation 2002-1: Quality Assurance for Safety Software at Department of Energy Nuclear Facilities*, Report, dated 3-13-03.

SQA Evaluator—an independent reviewer of the computer software, who is not affiliated with the software developing organization. It is required that the review organization or individuals have a thorough understanding of the applicable SQA requirements, expert level knowledge and application experience with the software in question, and an awareness of the overall context for the use of the subject software as part of the DOE safety process. The SQA evaluator is responsible for documenting the SQA evaluation of the candidate software, and based on this evaluation, confirms that the software SQA satisfactorily meets requirements for inclusion to the Central Registry.

DOE Office of Environment, Safety and Health, Office of Quality Assurance Programs— reviews the candidate software SQA evaluation and decides whether the candidate software should be included in the Central Registry.

Independence between the evaluator and the sponsor is critical for completion of a formal SQA evaluation, and should be maintained throughout the Central Registry submittal process. Ideally, the two participants should be based out of different organizations. In addition, while the SQA evaluator and the sponsor can be collocated at the same site, they should be functionally separated.

Before a software application and an independent evaluation are transmitted to DOE for consideration to the Central Registry, it is recommended that the software sponsor notify DOE Office of Quality Assurance Programs of its intentions. The notification will allow DOE to review usage characteristics of the software and the credentials of the designated evaluator. A screening review of this nature will minimize software and evaluation submittals that are not likely to be successful.

B.2 PROCESS

B.2.1 ADDING SOFTWARE APPLICATIONS TO THE CENTRAL REGISTRY

Submittal of a software application for consideration as toolbox-equivalent is a two-phase documentation effort, consisting of strategic benefits and SQA technical basis phases. In principle, the first phase should be prepared by the software sponsor, and needs to establish the basis or rationale for including the software in the Central Registry. At minimum, the discussion in the first phase should establish the following.

- Widespread use of the software across the DOE complex for safety related applications.
- Methods to ensure proper software information, error reporting, configuration control and other SQA management interface with the Central Registry.
- Demonstrated and quantifiable benefit for designating the software for the Central Registry.

The second phase, the SQA technical basis phase, is initiated with completion of an independent SQA evaluation, and is performed by the software evaluator. The evaluation should demonstrate satisfactory compliance with toolbox-equivalency criteria and requirements established for the

DOE Central Registry. The software sponsor is requested to provide information on the programs and procedures associated with the development, maintenance, and use of the subject software. An input template for this purpose has been developed, and is recommended as a starting point mechanism to solicit basic SQA information from the software sponsor. An electronic copy may be obtained from SQA Knowledge Portal under the SQA Library (*Software Information Template*), http://www.eh.doe.gov/sqa/doc_library.htm.

The input template seeks the following set of documents from the software developer.

1. Software project management and SQA plans
2. Software risk management documents
3. Software configuration management plan
4. Procurement and supplier management documents
5. Software requirements specifications
6. Software design, model description, programmer's reference, and related documents
7. Software design and related documents
8. Verification and validation, test report, and other documents
9. Software error notification and corrective action reports
10. User instructions, user manuals, and training packages/user qualification documents

Files, reports, telephone conferences, and other documented communications can provide confirmatory indications that actions have been performed in SQA, and these can be used in lieu of the availability of formal documents. However, formal documents are preferred because they explicitly demonstrate compliance with the primary criteria. Furthermore, formal documents reduce the uncertainty in verifying completion of an action.

Software practices discussed in Section 5 of this Guide and the corresponding documents for assessing compliance are listed in Table B-1, and are similar to those used in the evaluation of the initial software applications designated for the Central Registry. The details associated with each work activity are discussed in detail in the SQA plan and criteria document at the SQA Central Registry Web site.[3]

Because current and potential Central Registry software is best described under the custom developed category, requirements for evaluation of software should be consistent with the grading approach for custom developed software. Table B-2 lists SQA work activities discussed in Section 5 of this Guide for custom developed software at both A and B grading levels. Also shown is the SQA requirement from those used to evaluate the initial software applications designated for the Central Registry that best matches the DOE O 414.1C SQA work activity. For many of the SQA work activities, the match is only partial, (i.e., not all of a specific work practice is covered by the SQA toolbox requirement).

[3] U.S. Department of Energy, *Software Quality Assurance Plan and Criteria for the Safety Analysis Toolbox Codes*, Revision 1, dated 11-03.

Table B-1. Software Quality Assurance Work Activity and Corresponding Documentation for Demonstrating Compliance

DOE O 414.1C SQA Work Activity	SQA Documents
1. Software Project Management and Quality Planning	- Software Project Management Plan (SPMP) and/or - Software Quality Assurance Plan (SQAP) - Software Safety Plan
2. Software Risk Management	- Various document types can be used to cover risk management
3. Software Configuration Management	- Software Configuration Management Plan (SCMP) or related documents
4. Procurement and Supplier Management	- Contractual documents or other Software procurement and use agreement documentation
5. Software Requirements Identification and Management	- Software Requirements Specifications (SRS) or related document
6. Software Design and Implementation	- Software design description (SDD) Model Description, Programmer's Reference Manual, or other related documents
7. Software Safety	- SDD - Software Safety Analysis documentation
8. Verification and Validation	- Verification and Validation Report - Test Case Description and Outcome Report; Other testing documents
9. Problem Reporting and Corrective Action	- Software Error Notification and Corrective Action Report
10. Training of Personnel in the Design, Development, Use and Evaluation of Safety Software	- User Instructions or User Manuals - Training Packages and User Qualification

B.2.1.1 Evaluation Process

The SQA evaluator performs and documents a review of the software, using the inputs from the code developer, including the responses in the Software Input Template or the equivalent, and other communications. In cases where the software developer is unable to supply requested inputs, the SQA evaluation may consider alternative sources of information. Examples of alternative information are previous reviews,[4] older documentation from the code developer, technical and journal articles, and previous software comparison studies.

The size of the actual SQA evaluation effort, whether one individual or a team of subject matter experts, depends on the complexity of the software application. Regardless of SQA evaluation team size, those involved should be experienced in use of the software, but also knowledgeable of the evaluation criteria. It is recommended that the evaluation of the software work activities covered in Table B-1 use a sub-matrix of finer criteria to adequately evaluate the constituent parts of the requirement. Qualitative ranking of compliance was used with the designated toolbox software, applying the four terms defining compliance conditions: *Yes (meets requirement)*, *No (does not meet requirement), Uncertain (insufficient information available to evaluate),* and *Partial (some but not all criteria are met)*. Upon completion of the evaluation of each of the SQA work activities, the SQA evaluator can review results as a whole and render an overall assessment. The process leads to a firm basis to document findings in a verifiable, objective manner.

Table B-3 contains a procedure for evaluating toolbox-equivalent candidate software, defined in the custom developed category for most safety applications. The overall evaluation process is shown schematically in Figure B-1. Input information for the evaluation is based on receipt of a *Software Information Template.* An electronic copy may be obtained from SQA Knowledge Portal under the SQA Library (*Software Information Template*), http://www.eh.doe.gov/sqa/doc_library.htm.

While grading Level C cases can be postulated, it is believed that most software application candidates for the Central Registry are categorized best under grading Levels A and B.

The SQA evaluation (gap analysis) reports performed on the six initial toolbox codes are a reasonable level of detail for SQA evaluation documentation. While the SQA requirements and criteria used for the toolbox codes are similar to those described in this Guide, they differ in emphasis and extent of coverage. Thus, the gap analysis reports are illustrative, but not directly applicable models. Instead, a software evaluation template for this purpose has been developed.

The toolbox-equivalent software input and evaluation templates, as well as, copies of the gap analysis reports and the full SQA evaluation plan and criteria document, can be downloaded from the Central Registry Web site (http://www.eh.doe.gov/sqa/central_registry.htm).

[4] If previous reviews are used in whole or in part, it is required to confirm that the older review results are still applicable.

Table B-2. Software Quality Assurance (SQA) Requirements by Software Grading Level and Matching DOE O 414.1C SQA Work Activities

DOE O 414.1C SQA Work Activity*	Software Grading Level		Corresponding SQA Toolbox Software Requirement*
	Level A Custom	Level B Custom	
(a) Software project management & quality planning	Full**	Full	2. SQA Procedures and Plans
(b) Software risk management	Full	Graded***	Not addressed in the list of SQA requirements.
(c) Software configuration management	Full	Full	12. Configuration Control 14. Access Control
(d) Procurement and supplier management	Full	Full	3. Dedication
(e) Software requirements identification and management	Full	Full	5. Requirements
(f) Software design and implementation;	Full	Full	6. Design 7. Implementation
(g) Software safety	Full	Graded	6. Design
(h) Verification and validation	Full	Graded	8. Testing 10. Acceptance Test 11. Operation and Maintenance
(i) Problem reporting and corrective action	Full	Graded	13. Error Impact
(j) Training of personnel in the design, development, use and evaluation of safety software	Full	Full	9. User Instructions

*The SQA requirements used for evaluation of the initial set of software applications designated for the Central Registry are matched to the corresponding SQA work activity from DOE O 414.1C. See Table 3-3 of U.S. Department of Energy, *Software Quality Assurance Plan and Criteria for the Safety Analysis Toolbox Codes*, Revision 1, (November 2003) for details on the requirements and the labeling (numbering) scheme.

**Required for the computer software

***Graded depending on the application and based on judgment of SQA evaluator.

Table B-3. Plan for Evaluation of Candidate Software for Central Registry

Step	Procedure
1. Review Documentation	Determine that sufficient information is provided by the software developer to allow proper classification of the software. Review developer reports, previous evaluations, and conference and journal submittals, etc. Interview software developer.
2. Evaluate Justification (Rationale) for Including Software in Central Registry	Review software sponsor's document: Widespread use of the software across DOE complex for safety related applications? Methods to ensure proper software information, error reporting, configuration control and other SQA management interfaces with the Central Registry? Demonstrated and quantifiable benefit for designating the software for the Central Registry?
3. Process Software Information Template	Download template from Software Quality Assurance Web site, http://www.eh.doe.gov/sqa/doc_library.htm. Confirm graded level determination.
4. Assess Software Project Management and Software Quality Assurance Plans	Review software project management plan (SPMP) and software quality assurance plan (SQAP) for— • required activities, documents, and deliverables and • level and extent of reviews and approvals, including internal and independent review. Confirm that actions and deliverables (as specified in the SQAP) have been completed and are adequate. Review engineering documentation identified in the SPMP and SQAP, including— • software risk management documents; • software configuration management plan; • procurement and supplier management documents; • software requirements specifications; • software design, model description, programmer's reference, and related documents; • software design and related documents; • verification and validation, test report, and other documents; • software error notification and corrective action reports; and • user instructions, user manuals, and training packages/user qualification documents.

Step	Procedure
5. Assess SQA Work Activity	Review SQA documentation against detailed criteria found in the Software Evaluation Template for DOE O 414.1C SQA Work Activities: • software project management & quality planning, • software risk management, • software configuration management, • procurement and supplier management, • software requirements identification and management, • software design and implementation, • software safety, • verification and validation, • problem reporting and corrective action, and • training of personnel in the design, development, use and evaluation of safety software.
6. Document Evaluation Using Software Evaluation Template.	Use gap analysis reports as examples.

B.2.1.2 Submittal to the Central Registry

Once the SQA evaluation has been conducted and documented, the software may be submitted to the Central Registry under one of the following cases.

1. The software sponsor concludes in the software evaluation (gap analysis) that software has met all major requisite criteria in the ten SQA work activities, and no criterion is evaluated as "No (=not met)." In other words, all significant improvement actions are completed before the software is submitted for consideration as "toolbox-equivalent" to the Central Registry.

2. The software sponsor has identified one or more criteria not compliant for the subject software based on the gap analysis. However, the software sponsor can document a compelling technical basis for submitting the software as "toolbox-equivalent" to the Central Registry. Part of the technical basis should include a software application guidance report that points out specific limitations and weaknesses and provides instructions to the user on informed use of the subject software despite the identified gaps and other vulnerabilities. Examples of guidance reports prepared for the initial six codes designated for the Central Registry may be downloaded from the DOE SQA Web site at http://www.eh.doe.gov/sqa/doc_library.htm.

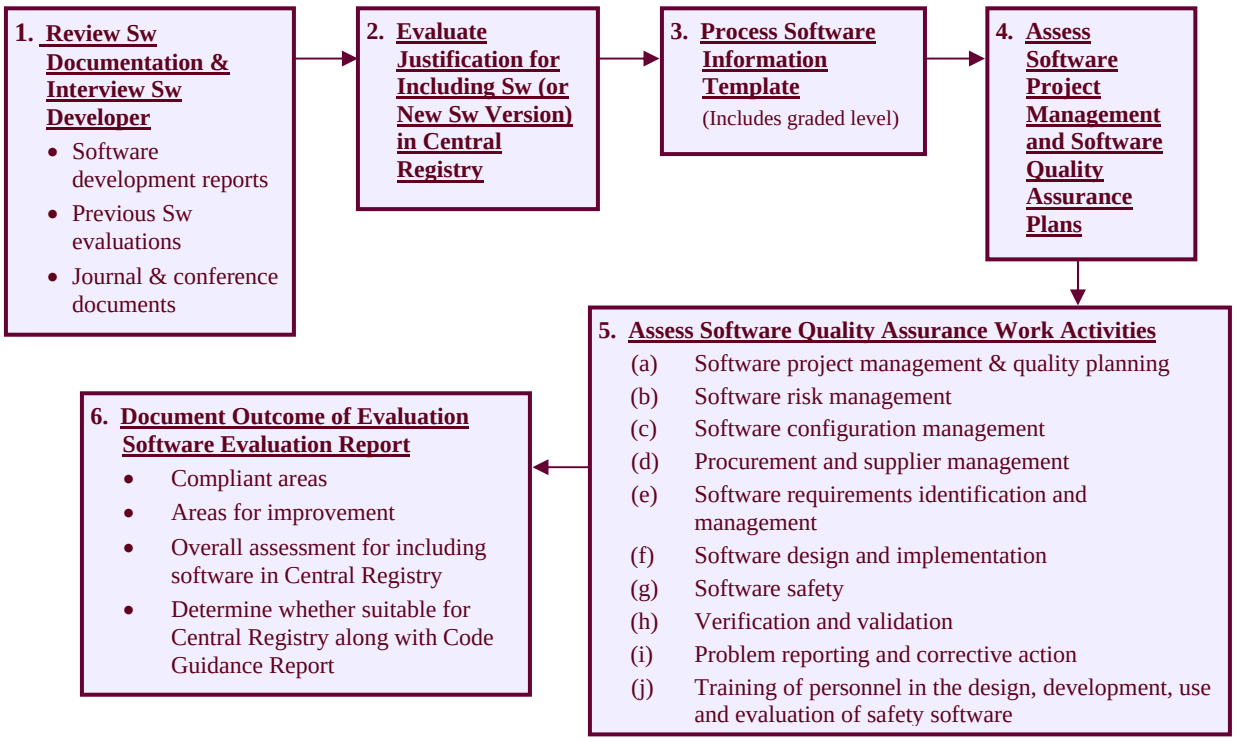


Figure B-1. Flow Sheet for Software Evaluation

If all substantive issues in either Case 1 or Case 2 are satisfactorily dispositioned, the software sponsor may move forward with the toolbox software submittal process. An electronic mail message should be sent to sqa@eh.doe.gov, requesting a review of the evaluation and designation of the software as a toolbox software application. All supporting documentation should be transmitted as attachments.

The DOE Office of Quality Assurance Programs will review the submittal in a timely manner. Table B-4 lists several of the key acceptance criteria for rendering a decision to include the candidate software in the Central Registry. A decision on designation of the candidate software as a toolbox software application will be communicated to the software developer and evaluator organizations. If the decision is favorable, the appropriate links will be provided for the software in question, and a general notice will be posted on the Central Registry Web site. Additional notification methods may be implemented to ensure broad notification of the changes in the Central Registry software collection.

If, on the other hand, issues with the subject software are irreconcilable, then the software sponsor is advised not to proceed further with the submittal process. It may be prudent to examine continued use of the software at the site in question, and explore use of alternative software, such as software currently contained in the Central Registry, for the specific safety application.

Table B-4. Primary Criteria for Deciding on Inclusion of Software to the Central Registry

Phase	Criterion*
1. Rationale for Adding Software to Central Registry	a. Widespread use of the software across DOE complex for safety related applications. b. Methods to ensure proper software information, error reporting, configuration control, and other software quality assurance (SQA) management interfaces with the Central Registry. c. Demonstrated and quantifiable benefit for designating the software to the Central Registry.
2. SQA Technical Basis	a. The SQA evaluation document adequately demonstrates that the candidate software has met all major requisite criteria, and no criterion is evaluated as "No (=not met)." If remedial tasks were cited before all criteria are considered met, it is determined that these have been completed. or b. The SQA evaluation document has identified one or more criteria not compliant for the subject software based on the gap analysis. However, a compelling technical basis is made for submitting the software as "toolbox-equivalent" to the Central Registry. Part of the technical basis should include a guidance report that points out specific limitations, weaknesses, and provides instructions to the user on informed use of the subject software despite identified gaps and other vulnerabilities.

*This is a partial list—others may be added as the process for software addition matures.

B.2.2 REVISIONS TO SOFTWARE APPLICATIONS IN THE CENTRAL REGISTRY

In the typical life-cycle processes associated with most software applications, updates, improvements, and modifications will be made. Similar to software that is being considered for the first time, revised software in the form of a new software version may also be submitted for inclusion in the Central Registry, with accompanying removal of the older version.

The same process is followed for revised software to be placed in the Central Registry as is outlined above for new software applications. The steps may be summarized as follows.

1. The software sponsor identifies the SQA evaluator organization.
2. The evaluator performs a complete evaluation over all aspects of the new software version, emphasizing new and revised aspects of the software application.
3. Upon conclusion of the evaluation and issuance of the SQA evaluation report (the gap analysis), the software sponsor decides whether software has satisfactorily met all requisite criteria for the ten SQA work activities, the revised software may be submitted to the Central Registry.

4. As noted earlier for new software applications to the Central Registry, an electronic mail message should be sent to sqa@eh.doe.gov, requesting a review of the evaluation and designation of the software as a toolbox software application. All supporting documentation should be transmitted as attachments.

5. The DOE Office of Environment, Safety and Health, Office of Quality Assurance Programs will review the submittal and decide on designation of the candidate software as a replacement version to existing toolbox software. Upon reaching a favorable determination, the appropriate links will be provided for the software version, and a general notice will be posted on the Central Registry Web site regarding a new software revision. In parallel with this action, the older software version will be removed from the Central Registry and designated as an "archived toolbox version."

B.2.3 REMOVAL OF SOFTWARE APPLICATIONS FROM THE CENTRAL REGISTRY

Software applications are also subject to being removed from the Central Registry. Several causes for this action include but are not limited to the following.

1. The software developer indicates that older versions will no longer be supported and elects to retire the software.

2. New survey information indicates that few if any sites are using the software and that other software applications are being used for the specified safety applications.

3. The DOE Office of Environment, Safety and Health, Office of Quality Assurance Programs may make a decision to formally remove the software due to accumulated evidence of unsatisfactory SQA events. Significant software errors in the subject software or other factors may lead to this outcome.

Regardless of the basis, the subject software application may be removed from the Central Registry after notification is posted on the Web site for a comment period of 60 days, and no compelling evidence is received that conflicts with the planned removal action. The notification should cite the basis or bases for the removal along with supporting documentation.

Upon reaching the end of comment period, the software application is then removed from the Central Registry and designated as an "archived software application."

B.2.4 ISSUE RESOLUTION AND ACTION COMMUNICATION

Actions will be taken on the addition, revision, and removal of software or when it is necessary to communicate information about software contained in the Central Registry. Several communication mechanisms may be used to alert DOE staff, DOE software user, and stakeholder groups. The extent of the communication will be commensurate to the action taken or the importance to safety of the issue, and will be decided by the DOE Office of Environment, Safety and Health, Office of Quality Assurance Programs.

Several of the primary mechanisms may include, but are not limited to, announcements to one or more of the following.

1. DOE Office of Environment, Safety and Health Central Registry Web site home page within the DOE Office of Environment, Safety and Health Knowledge Portal (http://www.eh.doe.gov/sqa/central_registry.htm)
2. Safety Analysis Working Group in EFCOG, particularly its Steering Committee, and its Accident Analysis Subgroup (http://www.efcog.org/workgroups/sawg-aa/)
3. Radiation Safety Information Computational Center (RSICC), (http://www-rsicc.ornl.gov/rsicc.html)
4. Use of the Central Registry e-mail distribution list
5. Formal Letter Notification to the Program Secretarial Officers

B.3 REFERENCES

1. American Society of Mechanical Engineers, Re: Comments on the Benefits of National Nuclear Quality Assurance Standards for NNSA and DOE Nuclear Activities and Oversight, Letter to Linton F. Brooks, NNSA (2002).
2. ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, American Society of Mechanical Engineers,* 2001.
3. ASME NQA-1A-1999, Addenda to ASME NQA-1-1997, *Quality Assurance Requirements for Nuclear Facility Applications,* American Society of Mechanical Engineers, 1999.
4. 10 CFR 830, Nuclear Safety Management.
5. 10 CFR 50, Appendix B, Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants.
6. DNFSB/TECH-25, *Quality Assurance for Safety-Related Software at Department of Energy Defense Nuclear Facilities,* Defense Nuclear Facilities Safety Board Technical Report, January 2000.
7. DNFSB Recommendation 2002-1, Quality Assurance for Safety-Related Software, Defense Nuclear Facilities Safety Board, September 2002.
8. International Organization for Standardization (ISO)9001-1994, *Quality Systems—Model for Quality Assurance in Design, Development, Production, Installation and Servicing,* ISO, 1994.
9. ISO 9001-2000, *Quality Management Systems—Requirements,* ISO 9000-3, ISO Quality management and quality assurance standards—Part 3: Guidelines for the application of

ISO 9001:1994 to the development, supply, installation and maintenance of computer software.

10. U.S. Department of Energy, Office of Environment, Safety and Health, Designation of Initial Safety Analysis Toolbox Codes, Memorandum to Linton Brooks, Defense Programs and Jessie Hill Roberson, Office of Environmental Management, March 28, 2003.

11. DOE-STD-3009-94, *Preparation Guide for U.S. Department of Energy Nonreactor Nuclear Facility Documented Safety Analyses,* dated 7-94.

APPENDIX C. USE OF ASME NQA-1-2000 AND SUPPORTING STANDARDS FOR COMPLIANCE WITH DOE 10 CFR 830 SUBPART A AND DOE O 414.1C AND SAFETY SOFTWARE

This appendix provides guidance on the use of ASME NQA-1-2000, *Quality Assurance Program for Nuclear Facilities,* and supporting standards for compliance with the Department of Energy's (DOE's) quality assurance (QA) requirements (10 CFR 830 Subpart A and DOE O 414.1C, *Quality Assurance,* dated 6-17-05) and their application to safety software.

C.1. PURPOSE

This guidance may be used by organizations adopting ASME NQA-1-2000 as a national consensus standard for development and implementation of a quality assurance program (QAP) that meets the DOE QA requirements and includes safety software within its scope. This appendix describes how ASME NQA-1-2000 addresses the DOE QA requirements and identifies DOE QA requirements that are not addressed by ASME NQA-1-2000. Selected standards from other standards bodies are included where emphasis or detail for safety software quality is necessary.

C.2. INTRODUCTION

DOE QA requirements for activities that affect, or may affect, quality, nuclear safety or other site specified criteria are established by 10 CFR Part 830 Subpart A, Quality Assurance Requirements. DOE also has equivalent requirements for all other federal and contractor activities in DOE O 414.1C. The DOE QA requirements and Guides are available for review at http://tis.eh.doe.gov/nsps/quality.html.

The DOE's objective of the QA Rule and Order is for organizations to establish effective integrated management systems (i.e., QAPs) for the performance of DOE nuclear-related work. The objective is accomplished through performance oriented QA criteria, coupled with appropriate technical standards to manage, perform, and assess work activities. The DOE Rule requires the use of voluntary consensus standards in the development and implementation of the QAP. The ASME NQA-1-2000 standard is a national consensus standard, and as indicated in DOE O 414.1C, ASME NQA-1-2000 or other national or international consensus standards that provide an equivalent level of quality assurance requirements as NQA-1-2000 must be used for providing the essential implementing methods for a QAP, including details for effective and reliable supporting processes and procedures, as presented in this subpart.

C.3. DOE RULE AND ORDER GENERAL ADMINISTRATIVE QAP REQUIREMENTS

The DOE Rule and Order include both administrative and regulatory quality requirements. Those administrative requirements relating to QAP approval authority, change control authority, and compliance should not be relevant to the scope of ASME NQA-1-2000. Other administrative quality related requirements that are relevant are addressed in Table C-1.

C.4. DOE RULE AND ORDER QA CRITERIA

The DOE Rule and Order include ten QA criteria that are used to develop and implement a QAP. Table C-2 identifies each of the ten DOE Rule and Order QA criterion and how they are addressed by the ASME NQA-1-2000, Part I requirements. Differences in the documents and topics that should be addressed independently of the ASME NQA-1-2000 criteria to meet the DOE criteria are described. In some cases, the ASME NQA-1 Part II, "QA Requirements for Nuclear Facility Applications," and Part IV, "Non-mandatory Guidance in ASME NQA-1-2000," are also appropriate to address the DOE requirements and describe *how* the QA criteria will be implemented. Table C-2 also includes selected standards from other standards bodies (IEEE and IAEA) where they add emphasis or detail for safety software quality.

TABLE C-1 10 CFR 830 Subpart A, dated January 10, 2001 §830.121 Quality Assurance Program DOE O 414.1C	
DOE General Requirements (Summarized)	**ASME NQA-1-2000 Requirements**
Graded Approach (830.7) Where appropriate, a contractor must use a graded approach to implement the requirements of this Part, document the basis of the graded approach used, and submit that documentation to DOE.	**Part I, Introduction, Requirement 1 and Requirement 2** provides for a graded approach to achieving quality by focusing on activities affecting quality and the application of requirements in a manner consistent with the relative importance of the item or activity. The cited text does allow for a graded approach, however a DOE QAP will need to describe how the graded approach is applied and documented to meet the DOE requirement. **Requirement 3, 801.4** provides grading relative to software. **Part II, Appendix 2A-2** Nonmandatory Guidance on Quality Assurance Programs includes guidance on this topic. **Part IV, 4.1, 101** **IAEA Technical Report (TR) Series 397, Appendix 1**
QAP Development & Implementation The QAP must describe how the DOE QA criteria are satisfied.	The ASME NQA-1 requirements partially meet the DOE requirement. **Requirement 2** requires that a documented QAP be planned, implemented and maintained; and requires the QAP provide for the planning and accomplishment of activities affecting quality. **Requirement 5** requires that "Activities affecting quality and services should be prescribed by and performed in accordance with documented instructions, procedures, or drawings that include or reference appropriate quantitative or qualitative acceptance criteria for determining that prescribed results have been satisfactorily attained." A DOE QAP will need to describe how the DOE criteria are satisfied.

TABLE C-1 10 CFR 830 Subpart A, dated January 10, 2001 §830.121 Quality Assurance Program DOE O 414.1C	
DOE General Requirements (Summarized)	**ASME NQA-1-2000 Requirements**
Integrated Management Systems The QA program must integrate the QA criteria with the Safety Management System (SMS) or describe how the QA criteria apply to the SMS.	The ASME NQA-1 requirements do not address the DOE requirement. A DOE QAP will need to address integration to meet the DOE criterion.
Ensuring Subcontractor & Supplier Quality The QAP must describe how the contractor responsible for the nuclear facility ensures that subcontractors and suppliers satisfy the QA criteria.	**Requirements 1, 2, 4, 7 and 18** The ASME NQA-1 requirements meet the DOE requirement by the establishment of quality interfaces between organizations, by the inclusion of applicable QA requirements in procurement documents, supplier evaluation activities and audits of suppliers. A DOE QAP will need to describe how subcontractors/suppliers satisfy the DOE criteria.

TABLE C-2 10 CFR 830 Subpart A, dated January 10, 2001 §830.122 Quality Assurance Criteria		
DOE Quality Assurance Criteria	**ASME NQA-1-2000 Requirements**	**Comments, Software Requirements & Other Standards (documents noted in bold italic)**
Criterion 1 - Management/Program	**NQA Requirements 1 and 2** The ASME NQA-1 requirements meet the DOE criterion, as noted.	***Part IV, 4.1, 400*** ***IEEE 730-2002*** ***IAEA TR 397, 2.2*** ***IAEA Nuclear Safety Guide (NS-G) NS-G-1.1, 4.11***
(1) Establish an organizational structure, functional responsibilities, levels of authority, and interfaces for those managing, performing, and assessing work.	The ASME NQA-1 requirements satisfy this element of the DOE criterion.	None
(2) Establish management processes, including planning, scheduling, and providing resources for the work.	NQA Requirement 1, 201 General and Requirement 2, 100 Basic meet the DOE criterion. ASME NQA-1 requires senior management to establish overall expectations for effective implementation of the quality assurance program and is responsible for obtaining the desired end result. This implies that adequate resources are provided to obtain desired results.	A DOE QAP will need to describe the management process for providing resources.

TABLE C-2 10 CFR 830 Subpart A, dated January 10, 2001 §830.122 Quality Assurance Criteria		
DOE Quality Assurance Criteria	**ASME NQA-1-2000 Requirements**	**Comments, Software Requirements & Other Standards (documents noted in bold italic)**
Criterion 2 - Management/Personnel Training and Qualification	**NQA Requirement 2** The ASME NQA-1 requirements meet the DOE criterion.	
(1) Train and qualify personnel to be capable of performing their assigned work. (2) Provide continuing training to personnel to maintain their job proficiency.	The ASME NQA-1 requirements satisfy these elements of the DOE criterion.	DOE-STD-1172-2003 Safety Software Quality Assurance Functional Area Qualification Standard ***IAEA TR 397, 2.4*** ***IAEA NS-G-1.1, 4.9&10***
Criterion 3 - Management/Quality Improvement	**NQA Requirements 2, 15, and 16** The ASME NQA-1 requirements partially meet the DOE criterion.	***Part II, 2.7, 204*** ***Part IV, 4.1, 204*** ***IAEA TR 397, 2.5*** A DOE QA Program will need to extend the requirements of ASME NQA-1 to ALL conditions adverse to quality not just significant conditions adverse to Quality.

TABLE C-2 10 CFR 830 Subpart A, dated January 10, 2001 §830.122 Quality Assurance Criteria		
DOE Quality Assurance Criteria	**ASME NQA-1-2000 Requirements**	**Comments, Software Requirements & Other Standards (documents noted in bold italic)**
(1) Establish and implement processes to detect and prevent quality problems.	The ASME NQA-1 requirements partially meet the DOE criterion. ASME NQA-1 provides a system of establishing quality requirements and monitoring compliance to prevent nonconforming conditions from causing quality problems. This is accomplished through various controls, inspections, and tests. Requirement 16 includes criteria to prevent recurrence of identified problems.	
(2) Identify, control, and correct items, services, and processes that do not meet established requirements.	The ASME NQA-1 requirements satisfy this element of the DOE criterion.	
(3) Identify the causes of problems and work to prevent recurrence as part of correcting the problem.	The ASME NQA-1 requirements partially satisfy this element of the DOE criterion for “significant” or “generic” nonconformances.	
(4) Review item characteristics, process implementation, and other quality-related information to identify items, services, and processes needing improvements.	The NQA requirements partially address this element of the DOE criterion for known deficiencies.	

TABLE C-2 10 CFR 830 Subpart A, dated January 10, 2001 §830.122 Quality Assurance Criteria		
DOE Quality Assurance Criteria	**ASME NQA-1-2000 Requirements**	**Comments, Software Requirements & Other Standards (documents noted in bold italic)**
Criterion 4 - Management/Documents and Records	**NQA Requirements 5, 6 and 17** The ASME NQA-1 requirements meet the DOE criterion.	
(1) Prepare, review, approve, issue, use, and revise documents to prescribe processes, specify requirements, or establish design. (2) Specify, prepare, review, approve, and maintain records.	The ASME NQA-1 requirements satisfy these elements of the DOE criterion.	***Part I, Requirement 3, 801*** ***Part II, 2.7, 201 & 802*** ***Part IV, 4.1, 201*** ***IAEA TR 397, 2.6 & 3.1*** ***IEEE 730, 829***
Criterion 5 - Performance/Work Processes	**NQA Requirements 5, 8, 9, 12, 13, and 14 and the Part I, Introduction** The ASME NQA-1 requirements meet the DOE criterion, as noted.	
(1) Perform work consistent with technical standards, administrative controls, and other hazard controls adopted to meet regulatory or contract requirements, using approved instructions, procedures, or other appropriate means.	The ASME NQA-1 requirements address "work" as activities affecting quality.	A DOE QA program will need to address "work" as broadly as the DOE criterion, since the requirements for "work" are derived from multiple sources in the DOE Rule and Order. ***Part I, Requirement 3, 802*** ***Part II, 2.7, 203 & 404***
(2) Identify and control items to ensure their proper use.	The ASME NQA-1 requirements satisfy this element of the DOE criterion.	

TABLE C-2 10 CFR 830 Subpart A, dated January 10, 2001 §830.122 Quality Assurance Criteria		
DOE Quality Assurance Criteria	**ASME NQA-1-2000 Requirements**	**Comments, Software Requirements & Other Standards (documents noted in bold italic)**
(3) Maintain items to prevent their damage, loss, or deterioration.	The ASME NQA-1 requirements satisfy this element of the DOE criterion.	***Part IV, 4.1, 203 & 405*** ***IAEA TR 397, 3.1 & 3.2*** ***IEEE 828-1998 & 1219-1998***
(4) Calibrate and maintain equipment used for process monitoring or data collection.	The ASME NQA-1 requirements satisfy this element of the DOE criterion.	
Criterion 6 - Performance/Design	**NQA Requirement 3** The ASME NQA-1 requirements meet the DOE criterion.	
(1) Design items and processes using sound engineering/scientific principles and appropriate standards. (2) Incorporate applicable requirements and design basis in design work and design changes. (3) Identify and control design interfaces. (4) Verify or validate the adequacy of design products using individuals or groups other than those who performed the work. (5) Verify or validate work before approval and implementation of the design.	The ASME NQA-1 requirements satisfy these elements of the DOE criterion.	***Part II, 2.7, 401 & 402*** ***Part IV, 4.1, 401 & 402*** ***ANS-10.4*** ***IAEA TR 397, 3.2 & 3.4*** ***IEEE 1012-1998 & 1012A-1998***

TABLE C-2 10 CFR 830 Subpart A, dated January 10, 2001 §830.122 Quality Assurance Criteria		
DOE Quality Assurance Criteria	**ASME NQA-1-2000 Requirements**	**Comments, Software Requirements & Other Standards (documents noted in bold italic)**
Criterion 7 - Performance/Procurement	**NQA Requirements 4 and 7** The ASME NQA-1 requirements meet the DOE criterion.	
(1) Procure items and services that meet established requirements and perform as specified. (2) Evaluate and select prospective suppliers on the basis of specified criteria. (3) Establish and implement processes to ensure that approved suppliers continue to provide acceptable items and services.	The ASME NQA-1 requirements satisfy these elements of the DOE criterion.	***Part II, 2.7, 300*** ***Part IV, 4.1, 300*** ***IAEA TR 397, 3.3***
Criterion 8 - Performance/Inspection and Acceptance Testing	**NQA Requirements 8, 10, 11, and 12** The ASME NQA-1 requirements meet the DOE criterion.	
(1) Inspect and test specified items, services, and processes using established acceptance and performance criteria. (2) Calibrate and maintain equipment used for inspections and tests.	The ASME NQA-1 requirements satisfy these elements of the DOE criterion.	***Part II, 2.7, 404*** ***Part IV, 4.1, 404*** ***ANS-10.4*** ***IAEA TR 397, 3.4*** ***IEEE 1008***

TABLE C-2 10 CFR 830 Subpart A, dated January 10, 2001 §830.122 Quality Assurance Criteria		
DOE Quality Assurance Criteria	**ASME NQA-1-2000 Requirements**	**Comments, Software Requirements & Other Standards (documents noted in bold italic)**
Criterion 9 - Assessment/Management Assessment	**NQA Requirement 2 and 18** The ASME NQA-1 requirements partially meet the DOE criterion, as noted	
Ensure managers assess their management processes and identify and correct problems that hinder the organization from achieving its objectives.	While ASME NQA-1, Requirement 2, 100 Basic, requires management to regularly assess the adequacy and effective implementation of quality assurance, the DOE criterion is broader in scope and intent.	***Part II, 2.7, 202*** ***Part IV, 4.1, 202*** ***IAEA TR 397, 4.1*** ***IEEE 1028-1997*** While audits per Req. 18 of NQA provide an input to this requirement, a DOE QAP will need to align with the intent, focus, and concepts described in DOE G 414.1-1A, *Management Assessment and Independent Assessment Requirements of 10 CFR 830.120 and DOE O-414.1, Quality Assurance*, to meet the DOE criterion.

TABLE C-2 10 CFR 830 Subpart A, dated January 10, 2001 §830.122 Quality Assurance Criteria		
DOE Quality Assurance Criteria	**ASME NQA-1-2000 Requirements**	**Comments, Software Requirements & Other Standards** **(documents noted in bold italic)**
Criterion 10 - Assessment/Independent Assessment	**NQA Requirements 1, 2, 10, 11, 15, 16, and 18** The ASME NQA-1 requirements meet the DOE criterion.	
(1) Plan and conduct independent assessments to measure item and service quality, to measure the adequacy of work performance, and to promote improvement. (2) Establish sufficient authority, and freedom from line management, for the group performing independent assessments. (3) Ensure persons who perform independent assessments are technically qualified and knowledgeable in the areas to be assessed.	DOE defines assessment as a general term that includes a variety of evaluation methods (i.e.; reviewing, evaluating, inspecting, testing, checking, surveillance, auditing, or otherwise determining and documenting). As such, several ASME NQA-1 requirements may be necessary to address the various DOE independent assessment methods. These activities when combined with the NQA corrective action requirement have the intent of the DOE criterion, to "promote improvement."	***IAEA TR 397, 4.2*** Assessment as a DOE activity for a DOE QAP will need to align with the intent, focus and concepts described in ***DOE G-414.1-1A, Management Assessment and Independent Assessment Requirements of 10 CFR 830.120 and DOE- O-414.1 Quality Assurance.***

APPENDIX D. QUALITY ASSURANCE STANDARDS FOR SAFETY SOFTWARE IN DEPARTMENT OF ENERGY NUCLEAR FACILITIES

D.1. INTRODUCTION AND REGULATORY BASIS

The Department of Energy (DOE) nuclear safety regulation, 10 CFR 830 Subpart A, establishes quality assurance requirements for activities, including providing items or services that affect or may affect nuclear safety of DOE nuclear facilities. The Quality Assurance (QA) Rule includes a requirement that consensus standards be used to develop and implement QA Programs. Safety software is included in the scope of activities covered by the QA Rule. Therefore consensus standards should be used for applying QA to safety software activities where practicable and consistent with contractual and regulatory requirements. This appendix describes practicable standards for safety software QA that may be used to satisfy the QA Rule.

D.2. REGULATORY AND QA PROGRAM COMPLIANCE

The ultimate responsibility for complying with the QA Rule, and for selecting standards for safety software that falls under the scope of the QA Rule, rests with the nuclear facility contractor. Nuclear facility contractors with DOE-approved QA Programs should ensure that any changes to their QA Program are made in accordance with the QA Rule and any supplemental DOE direction provided through contractual means.

D.3. QA PROGRAM STANDARDS VERSUS SOFTWARE STANDARDS

Dozens of consensus standards have been developed that address every aspect of software. In the broadest sense of QA, all of these standards could be interpreted as “QA standards.” To develop a useful report, it is necessary to limit discussion of standards to those that directly support compliance with the DOE QA Rule and development of a QA Program that includes safety software. There are other documents (e.g., technical reports, agency directives, and industry guides) that may be useful as examples for application of the standards, but they are not developed through an accredited consensus standards process.

D.4. STANDARDS USE IN A QA PROGRAM CONTEXT

Many of the standards developed address specific phases of software development rather than a QA program that encompasses safety software. In some cases the standards do cover a single criterion within the QA program, such as training. Where this type of standard is used, it should be in the context of the broader QA program that includes all criteria necessary for effective QA. This report will differentiate between QA program standards and standards that address a specific criterion.

D.5. QA PROGRAM AND SOFTWARE QUALITY STANDARD REQUIREMENTS

Identification of QA program standards for safety software should consider the following:

- compatibility with the DOE QA Rule;
- relevance to nuclear facility safety;
- applicability to software developed in-house, purchased, or modified;
- applicable to the entire software life-cycle; and
- inclusion of commonly accepted work activities for software QA.

D.6. NATIONAL STANDARD FOR NUCLEAR FACILITY QUALITY AND SOFTWARE

The most comprehensive nuclear QA program standard for application to safety software is the American Society of Mechanical Engineers ASME NQA-1-2000, *Quality Assurance Program for Nuclear Facilities*. Appendix C of this Guide includes SQA requirements that are compatible with the DOE QA Rule, can be integrated/supplemented with other standards, and is directly applicable to safety software. Most importantly, ASME NQA-1-2000 expands upon the DOE QA program requirements to specifically address requirements for software quality, thus, placing safety software quality in the context of the overall QA program. These specific software quality requirements are discussed in—

- ASME NQA-1, Part I, Requirement 3, Section 800, *Design Control*;
- Part II, Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications; and
- Part IV, Subpart 4.1, Guide on Quality Assurance Requirements for Software.

ASME NQA-1-2000 with Subpart 2.7 is also a practicable choice for implementing the DOE QA Rule for safety software because it—

- is easily supplemented with other IAEA, IEC, and IEEE standards;
- provides independence for development and verification;
- supports graded implementation;
- is widely used among DOE contractor QA programs; and
- is accredited as the American National Standard for nuclear application.

Table C-1 in Appendix C of this Guide describes how ASME NQA-1-2000 aligns with DOE QA criterion and includes other standards that further expand the content of ASME NQA-1 requirements for safety software.

Institute of Electrical and Electronic Engineers (IEEE) Std 7-4.3.2-2003[1], Criteria for Digital Computers in Safety Systems for Nuclear Power Generating Stations, describes computer specific requirements addressing firmware, software, and hardware alike for the development process in an integrated approach. This standard recommends a minimum set of functional and design requirements for computer components of a safety system employed in nuclear power generating stations. Additionally IEEE Std 1228, Standard for Software Safety Plans provides requirements for the development of a management plan and performance of safety software activities.

American Nuclear Society (ANS) standard, ANSI/ANS-10.4-1987[2] is supplemental to the IEEE Std 7-4.3.2-2003 since it targets activities to improve the reliability of scientific and engineering computer applications while mitigating the risk of incorrect applications. Additionally IEEE has a complete series of software and systems engineering standards to provide detail requirements and guidance for the development of safety software.

D.7. INTERNATIONAL STANDARDS FOR QUALITY AND SOFTWARE

D.7.1 INTERNATIONAL ATOMIC ENERGY AGENCY (IAEA)

The responsibility for international standards for nuclear safety is assigned to the International Atomic Energy Agency (IAEA). The IAEA has a significant number of standards, guides, and requirements for all aspects of nuclear facility safety, including software. The requirements and guidance for nuclear facility quality are addressed in a 1996 Safety Series "Code" No. 50-C-Q, *Quality Assurance for Safety in Nuclear Power Plants and other Nuclear Installations*, and Safety Guides 50-SG-Q1–Q14, respectively. The IAEA Code quality requirements closely parallel the DOE QA Rule.

IAEA safety software guidance is detailed in Technical Reports (TR) Series No. 397, *Quality Assurance for Software Important to Safety*. This TR provides information and guidance for defining and implementing QA programs covering the entire life-cycle of software important to safety. TR 397 was developed using a large amount of available information and standards and offers implementation guidance that is tied to the QA program requirements found in the IAEA Code. The application guides are useful aids for developing QA programs for safety software, specifically:

- Appendix I, illustration of a graded software quality assurance program;
- Appendix III, considerations before acquisition of computerized tools;
- Appendix IV, functions of computer program understanding and reverse engineering tools;

[1] IEEE Std 7-4.3.2-2003, *IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations*, Institute of Electrical and Electronic Engineers, 2003.
[2] American National Standards Institute (ANSI)/American Nuclear Society (ANS) 10.4-1987 (R1998), *Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry*, ANS, 1998.

- Appendix V & VI, general training guideline and proposed outlines for training;
- Appendix VII, characteristics of defect prevention process;
- Appendix VIII, examples of software development life-cycle models;
- Appendix IX, recommendations for design input documentation for monitoring, control, and safety system software;
- Appendix X, recommendations for software development plans applicable to monitoring, control, and safety system software;
- Appendix XI, recommendations for standards and procedures handbooks applicable to monitoring, control, and safety system software;
- Appendix XII, recommendations on the content of software requirements specifications for monitoring, control, and safety system software;
- Appendix XIII, recommendations on software design descriptions for monitoring, control, and safety system software;
- Appendix XIV, recommendations on design and development documents for design, engineering, and analysis software;
- Appendix XV, recommendations on application documents for design, engineering, and analysis software;
- Appendix XVI, suggested good coding practices for design, engineering, and analysis software;
- Appendix XVII, recommendations on programming of monitoring, control, and safety system software;
- Appendix XVIII, discussion of verification and validation methods;
- Appendix XIX, recommendations on verification reports and activities for monitoring, control, and safety system software; and
- Appendix XX, recommendations on commissioning monitoring, control, and safety system software.

TR 397, and IAEA Safety Guide (SG) Series No. NS-G-1.1, *Software for Computer Based Systems Important to Safety in Nuclear Power Plants*, provide expanded information that can be fully integrated with the ASME NQA-1-2000 requirements and the DOE QA Rule to produce an effective quality program for safety software. Relevant portions of TR 397 are referenced in Appendix C of this Guide to illustrate their relationship to the DOE QA Rule criteria and ASME NQA-1 requirements.

D.7.2 INTERNATIONAL ELECTROTECHNICAL COMMISSION (IEC)

The IEC is responsible for several software standards in the nuclear power plant arena. These standards are referenced in the IAEA TR 397. Those standards include IEC 880 Software for Computers in the Safety Systems of Nuclear Power Stations, IEC 987 Programmed Digital

Computers Important to Safety for Nuclear Power Stations, IEC 1226 Nuclear Power Plants—Instrumentation and Control Systems Important for Safety—classification, IEC 61508 Parts 1-7 Functional Safety of Electrical/Electronic/Programmable Electronic Safety Related Systems and, IEC 61511 Parts 1-3 Functional Safety – Safety Instrumented Systems for the Process Industry Sector.

The IEC Standard 880[3] is applicable to Level A highly reliable safety systems of nuclear power plants. Like its Canadian counterpart, CE-1001-STD, IEC 880 standard advises various approaches to maximize the reliability of the safety systems within a nuclear power plant.

D.7.3 INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO)

The International Organization for Standardization (ISO) is responsible for ISO 9001-2000, *Quality management systems – requirements.* The ISO 9001 standard is designed for use internally or as a contractual requirement for generic quality systems. ISO 9001 does not specifically address computer software. More importantly, ISO is not chartered to develop standards for nuclear safety applications (this is the domain of the IAEA) and consequently lacks sufficient focus (and rigor) to address DOE nuclear facility hazards. Commercial industries that face high hazards and high mission/political risk similar to DOE (e.g., aerospace, telecom, chemical) have each issued supplemental requirements to improve on ISO 9001 for application to their industry.

Although ISO has a guide for applying a previous version of ISO 9001 (1994) to software (ISO 9000-3, ISO Quality management and quality assurance standards—Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software), this guide is not focused on nuclear safety.

Given that (1) the ISO standards are not developed for nuclear facility applications, (2) the IAEA is the internationally chartered standards body for that subject, and (3) the IAEA offers safety software quality standards compatible to DOE and ASME NQA-1, ISO should not be considered a practicable choice for standards in this subject area.

D.7.4 ATOMIC ENERGY CANADA LTD (AECL)

The Canadian standard[4] CE-1001-STD specifically recommends a minimum set of processes for the software quality engineering of "safety critical systems used in real-time protective, control, and monitoring systems" of Level A applications. This standard recommends particular detailed outputs for the software life-cycle processes, but is not prescriptive in how the outputs should be obtained.[5]

---

[3] IEC 880, *Software for Computers in the Safety Systems of Nuclear Power Stations,* International Electrotechnical Commission, 1986.
[4] CE-1001-STD, Rev. 2, *Standard for Software Engineering of Safety Critical Software,* CANDU Computer Systems Engineering Centre for Excellence, Atomic Energy of Canada Ltd. and Ontario Power Generation, Inc., 1999.
[5] Herrmann, Debra S., *Software Safety and Reliability: Techniques, Approaches, and Standards of Key Industrial Sectors,* IEEE Computer Society, 2000.

D.8. EXAMPLE APPLICATION GUIDES, FEDERAL AGENCY REQUIREMENTS AND PROCEDURES

D.8.1 DEPARTMENT OF DEFENSE (DOD)

The DoD software project requirement is Directive 5000.61 and related guidance. These documents address software development, verification, validation, accreditation, maintenance, review, and management. The documents also refer to national and industry standards. For example, independent review is addressed in the *Verification, Validation and Accreditation (VV&A) Recommended Practices Guide*. The guide also describes methods for assuring software using a graded approach depending on whether the software has—

- been previously accredited based on verification and validation data which is available;
- been previously accredited based on historical use;
- not been previously accredited, but some verification and validation data available; or
- not been previously accredited, with little or no verification and validation available.

DoD MIL-STD-882D[6] Appendix A is particularly useful because it supplies guidance for implementing a system safety effort, and the definitions, roles and responsibilities for an organization undertaking a new system safety effort. Similarly, the NASA Standard "Software Safety NASA Technical Standard"[7] provides general guidance for a software safety effort.

D.8.2 NATIONAL AERONAUTICS AND SPACE ADMINISTRATION (NASA)

The NASA software document is the NASA 8739.8, *Standard for Software Assurance*. Also reference NASA STD 8719.13B, *Software Safety*; and NASA-GB-8719.13 *NASA Software Safety Guidebook*. The NASA standard includes processes to establish and implement requirements and procedures, as well as, evaluating software products against requirements standards and procedures.

D.8.3 ENVIRONMENTAL PROTECTION AGENCY (EPA)

The EPA uses at least two standards for software in environmental safety projects. EPA regulates the DOE Waste Isolation Pilot Project (WIPP) using 40 CFR 194. The 40 CFR 194 Rule and the WIPP QA program requirements influence many other waste generation sites across the DOE complex. The regulation adopts ASME NQA-1, 1989, and NQA-2A-1990 Part 2.7. EPA also contracts for cleanup of certain Superfund sites. For these projects EPA has used the national standard *Quality Systems for Environmental Data and Technology Programs - Requirements with Guidance for Use*, ANSI/ASQC E4-1994. This standard is currently undergoing revision and includes requirements for software quality that parallel ASME NQA-1-2000. The standard also parallels the DOE QA Rule criterion.

---

[6] DoD MIL-STD-882D, *Standard Practice for System Safety*, U.S. Department of Defense, dated 2-10-00.
[7] NASA-STD-8719.13A, *Software Safety*, National Aeronautics and Space Administration, 1997.

D.8.4 DOE PROGRAM REQUIREMENTS AND PROCEDURES

The Department and its contractors have a variety of program requirement documents and implementing procedures for safety software in use for nuclear facilities. However, the Yucca Mountain Project's quality assurance requirements document (QARD) DOE/RW-0333P has been evaluated by an external regulatory body and found acceptable. The QARD and software quality supplements describe a rigorous graded approach to safety software suitable for review by other DOE organizations for use in developing their QA programs for safety software.

D.9. PRACTICABLE STANDARDS FOR DOE QA RULE IMPLEMENTATION

D.9.1 QA RULE & STANDARDS ALIGNMENT

The tables in Appendix C describe how ASME NQA-1-2000 aligns with DOE QA criterion. It also includes other standards that further expand the content of ASME NQA-1 requirements for safety software to address appropriate elements for safety software quality.

D.9.2 STANDARDS LISTING

Appendix G of this Guide contains a listing of standards that may be applied to safety software to ensure quality.

D.10. REFERENCES

1. ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, American Society of Mechanical Engineers,* 2001.

2. 10 CFR 830, Nuclear Safety Management.

3. 10 CFR 63, Disposal of High-Level Radioactive Wastes in A Geologic Repository at Yucca Mountain, Nevada.

4. DNFSB/TECH-25, *Quality Assurance for Safety-Related Software at Department of Energy Defense Nuclear Facilities,* Defense Nuclear Facilities Safety Board Technical Report, January 2000.

5. DNFSB/TECH-31 *Engineering Quality into Safety Systems,* Defense Nuclear Facilities Safety Board Technical Report, March 2001.

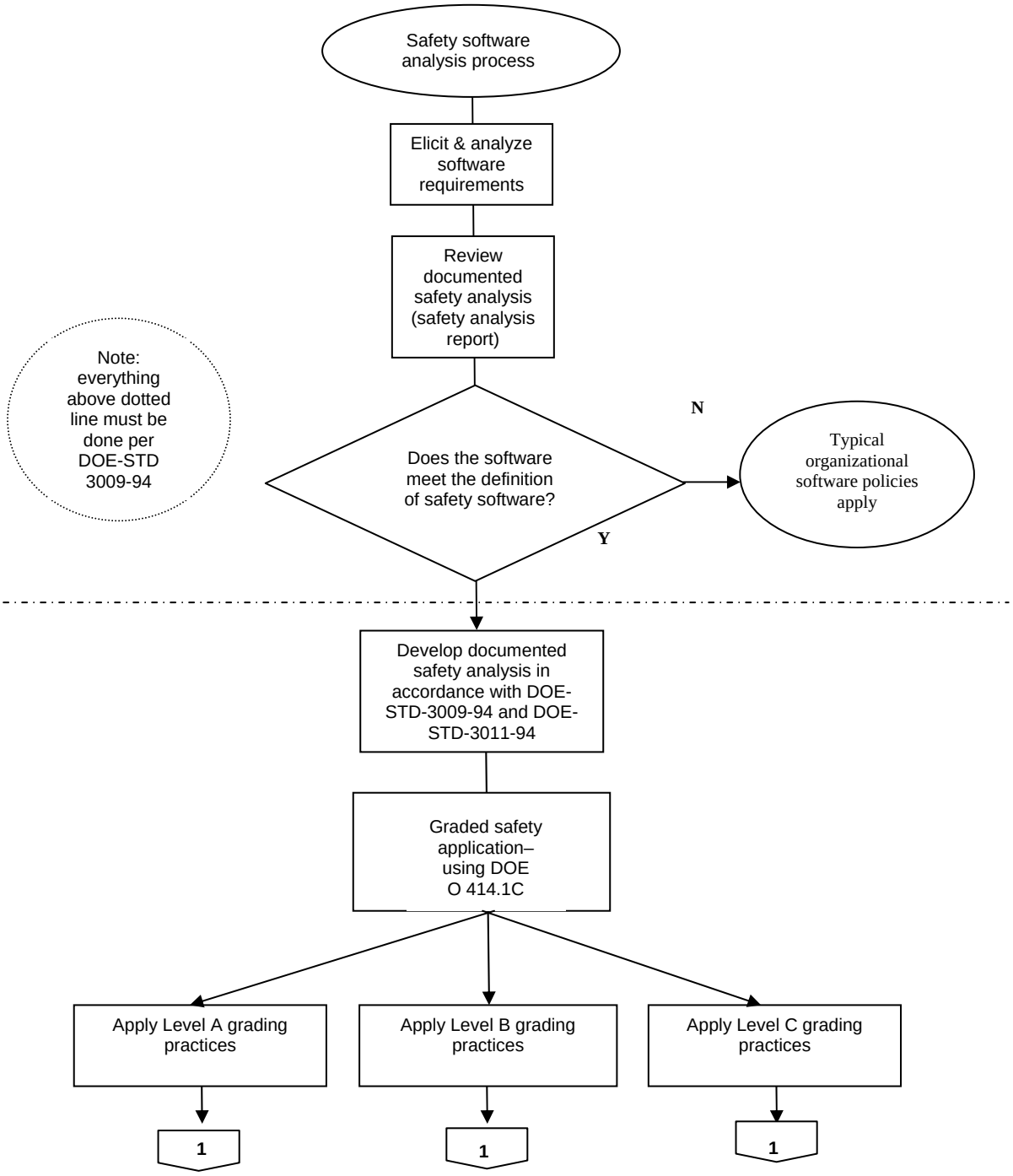
6. DNFSB Recommendation 2002-1, Quality Assurance for Safety-Related Software, Defense Nuclear Facilities Safety Board, September 2002.

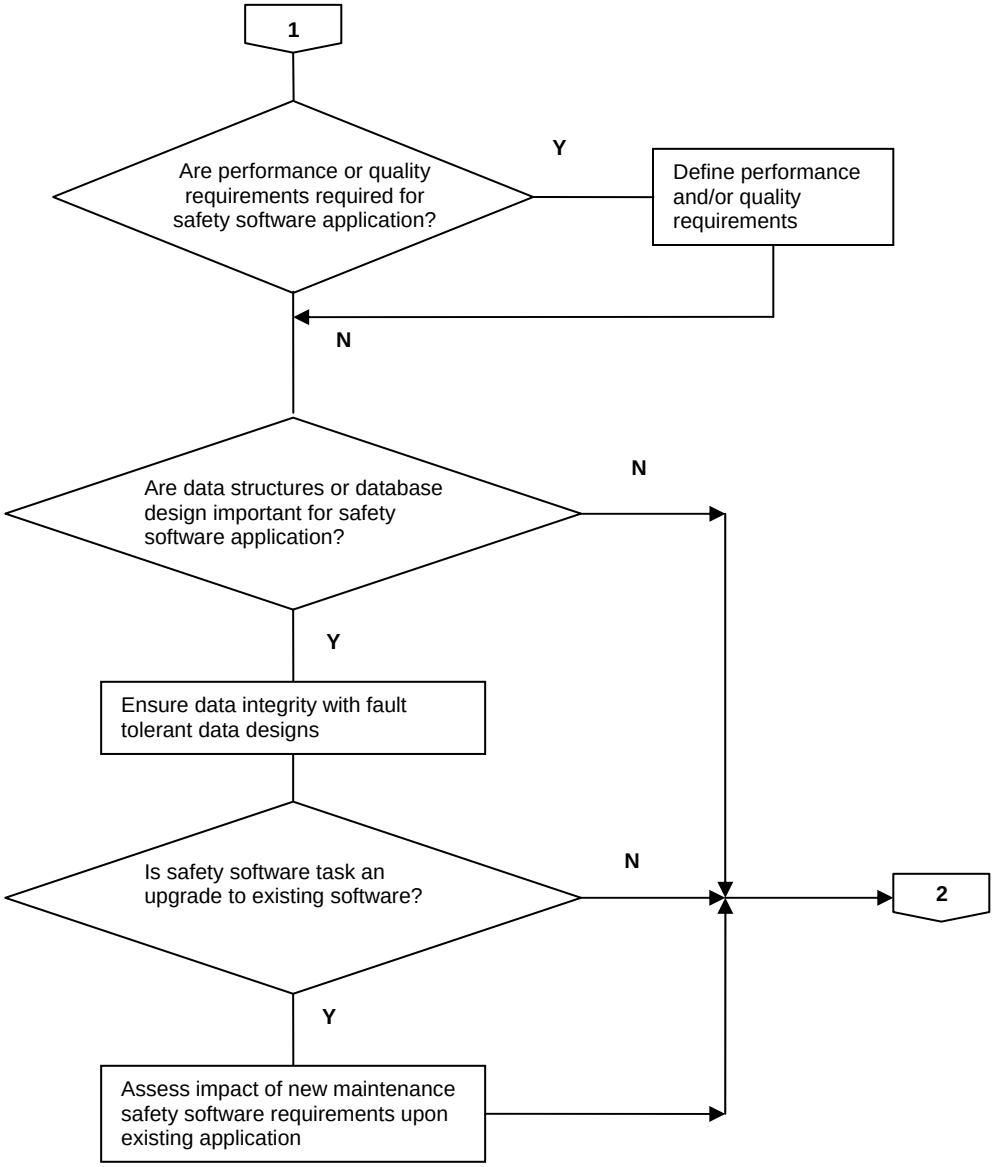
7. DOE Letter and Report, Quality Assurance for Safety-Related Software at Department of Energy Defense Nuclear Facilities, DOE Response to TECH-25, U.S. Department of Energy, October 2000.

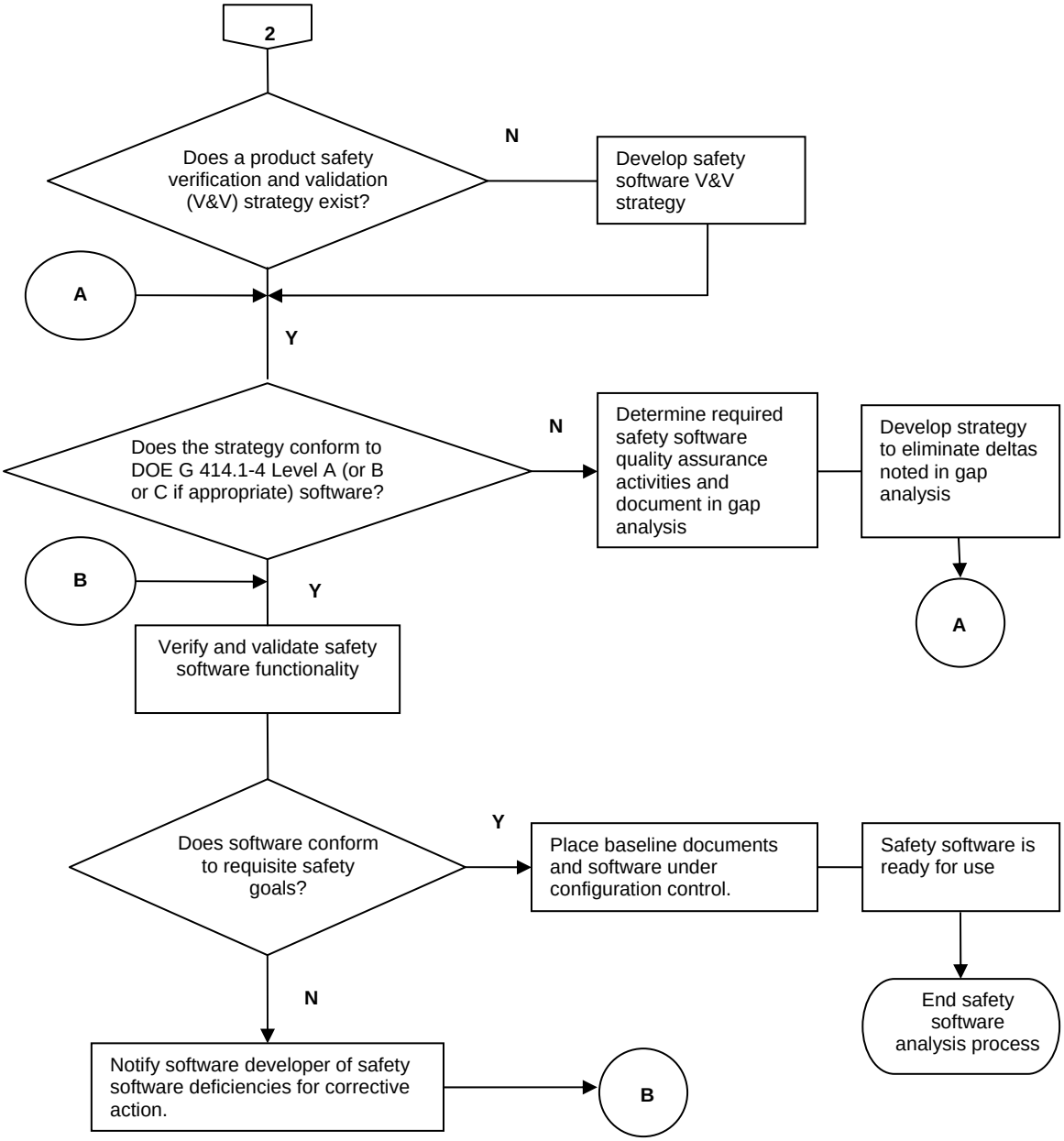
8. DOE Report, Implementation Plan for Defense Nuclear Facilities Safety Board Recommendation 2002-1: Quality Assurance for Safety Software at Department of Energy Nuclear Facilities, U.S. Department of Energy, March 13, 2003.

APPENDIX E. SAFETY SOFTWARE ANALYSIS AND MANAGEMENT PROCESS

The following diagrams provide a recommended process flow for the analysis and management of safety software applications.




1
Are performance or quality requirements required for safety software application?
Y
Define performance and/or quality requirements
N
Are data structures or database design important for safety software application?
N
Y
Ensure data integrity with fault tolerant data designs
Is safety software task an upgrade to existing software?
N
2
Y
Assess impact of new maintenance safety software requirements upon existing application


2
Does a product safety verification and validation (V&V) strategy exist?
N
Develop safety software V&V strategy
A
Y
Does the strategy conform to DOE G 414.1-4 Level A (or B or C if appropriate) software?
N
Determine required safety software quality assurance activities and document in gap analysis
Develop strategy to eliminate deltas noted in gap analysis
A
B
Y
Verify and validate safety software functionality
Does software conform to requisite safety goals?
Y
Place baseline documents and software under configuration control.
Safety software is ready for use
End safety software analysis process
N
Notify software developer of safety software deficiencies for corrective action.
B

APPENDIX F. DOE O 414.1C CRITERIA REVIEW AND APPROACH DOCUMENT

CONTENTS

DOE O 414.1C CRITERIA REVIEW AND APPROACH DOCUMENT

F.1 INTRODUCTION

This document contains software qualification assessment criteria and guidelines for assessing the safety system software used in Department of Energy (DOE) defense nuclear facilities.

This document is organized as follows.

- The ***Assessment Guidelines*** section covers the purpose, scope, guiding principles, and assessment methodology for assessing the processes currently in use for ensuring the adequacy of safety software.
- The ***Criteria and Approach*** section presents the objective, criteria, approach, and tailoring for the following work activities: (1) Software Project Management and Quality Planning, (2) Software Risk Management, (3) Software Configuration Management (SCM), (4) Software Procurement and Supplier Management, (5) Software Requirements Identification and Management, (6) Software Design and Implementation, (7) Software Safety, (8) Software Verification and Validation, (9) Software Problem Reporting and Corrective Action, (10) Training of Personnel in the Design, Development, Use, and Evaluation of Safety Software.
- The ***Report Format*** section provides a suggested report format.
- The ***References*** section lists selected references relevant to software quality assurance (SQA).

F.2 PURPOSE AND SCOPE

The purpose and scope of the criteria review and approach document (CRAD) is to provide a set of consistent criteria and guidelines for the assessment of the safety software. The safety software includes safety analysis; design of structures, systems, and components (SSCs); human-machine interface software; network interface software; programmable logic controller (PLC) programming language software; and safety management software in DOE defense nuclear facilities. DOE O 414.1C includes the definitions for safety software.

The assessment criteria and guidelines ensure that the software being used in DOE's nuclear facilities is adequate. The primary set of baseline SQA criteria for evaluating safety software are based on the following:

- 10 CFR 830, Nuclear Safety Management;
- DOE O 414.1C, *Quality Assurance*, dated 6-17-05;
- American Society of Mechanical Engineers (ASME) NQA-1-2000, *Quality Assurance Program for Nuclear Facilities;* and
- applicable Institute of Electrical and Electronics Engineers (IEEE) standards.

The CRAD is not intended to be prescriptive enough to evaluate the overall QA program associated with the safety software but rather to focus on the safety software application/product. Individual sites should tailor the scope of the assessment to suit the specific usage software in their safety systems. The CRAD could be used for assessment of the following types of software:

- custom software developed by DOE, its contractors, or subcontractors for use with safety systems or safety class (SC) SSCs;
- configurable, such as PLCs;
- acquired software, including commercial off-the-shelf (COTS) software;
- utility calculation software, such as spreadsheets and math programs (along with their associated user files), used to perform safety analysis and design calculations; and
- commercial design and analysis tools, such as proprietary facility design and accident analysis software.

These software types are used in the following safety software applications:

- safety management and administrative database programs and associated user files to maintain control of information that has nuclear safety implications;
- instrumentation and control software, including embedded microprocessors, distributed control systems, supervisory control and data acquisition systems, PLCs, and other related software;
- networking and interface applications;
- safety accident analysis; and
- design and analysis of SSCs.

Should an issue arise that casts doubt on the validity of software previously used to support design or development, it should be resolved using the unreviewed safety question (USQ) determination (USQD) process. Generic USQs will be used to the extent possible to preclude multiple facilities' developing separate USQDs for the same issue.

F.3. GUIDING PRINCIPLES

The following principles should guide the conduct of the assessment. The assessment team leader, with assistance from the DOE site manager responsible for these assessments, should ensure that these guiding principles are incorporated in the tailoring process for assessing safety software applications.

- The team should review any previous assessments and reviews of safety software. This review will enable the team to understand previous assessments, software qualification processes, associated requirements and performance criteria, assumptions concerning system operations, and the role of safety software in operations.

- The team should review any lessons learned from past events associated with software applications and include any additional attributes as appropriate in the assessment plan.

- The review of SQA processes for existing safety software should follow the guidance provided in the DOE G 414.1-4 Section 3.3.2 Existing Safety Software Applications.

- The physical boundaries of the software within the safety system or subsystem level or portions thereof under review should be agreed upon by DOE, the contractor line management, and the assessment team lead prior to the start of the assessment and should be documented in the assessment report.

- Care should be taken to balance the effort invested during the assessment in verifying the SQA processes and their supporting documentation, against the demonstrated effect on improving the software quality and safety, and on eliminating the costly errors that result from misunderstood requirements.

- The assessment of specific software applications should begin with gaining an understanding of the overall system, and documenting the system safety functions, the performance criteria that the system should meet to successfully accomplish its safety functions, and the role of the software in ensuring that these functions and criteria are met. The potential consequences of failure of the software and the associated effects on system operability should be understood and documented.

- The facility staff should assist the team in understanding the associated SQA process; provide documented evidence to the team that the appropriate SQA standards were applied to software development, procurement, or use; and provide a staff point of contact for further information.

- Procedures and records for software design, implementation, procurement, verification and validation (V&V), testing, and maintenance should be evaluated for adequacy and to determine whether they are appropriate and are being used to verify that software requirements and performance criteria described in the software requirements documentation are satisfied.

- If the team identifies a condition that poses an imminent threat to personnel or facility safety, line management should be notified immediately. Team personnel should immediately point out the imminent threat condition to their points of contact or appropriate facility manager and notify the assessment team leader as soon as practical.

- In some cases it will be necessary to tailor the assessment criteria and guidelines to focus the assessment to address those aspects determined to be appropriate for the assessment scope. The tailoring process is intended to ensure that the assessments are conducted in accordance with the criteria and guidelines that are appropriate and applicable to each specific situation. These assessment criteria and guidelines are intended to be flexible, and may be tailored to allow the most cost-effective use of resources in determining the operational readiness of safety software and its ability to operate safely on a continued basis. The tailoring process may take into account considerations, such as recently

completed assessments, evaluations, studies, inspections, and other relevant factors. The assessment criteria and guidelines in this CRAD are provided as a tool for use in developing specific criteria and guidelines. It is recognized that some of the criteria may not apply. This should be noted in the assessment report. For each assessment, the tailoring and its associated rationale should be agreed upon prior to the start of the assessment, and documented in the assessment report.

- The team should consider the level of modification to the software when evaluating the adequacy of the SQA processes. Acquired software, such as COTS, may not be modified and can be viewed within the system as a "black box." Custom developed software is completely modifiable and may require additional SQA processes over those of acquired software. Some acquired software can be configured specifically for its application or its source code can be modified to meet application specific requirements. In these instances a higher level of SQA requirements should be expected. However, these requirements may not be as high as custom developed software for the specific application. The grading approach in this Guide assists in this effort.

- The assessment should consider the effectiveness of SQA processes that are separate from system quality processes. In many instances, especially with acquired software, the separation of software from the system may increase costs but not increase the safe operation of the system.

- Information for existing software may not be appropriately documented. The team should determine whether any of the documentation, such as a problem statement, requirements specification, design specification, test plan, or test results, is available. In situations where clearly identifiable formal documents do not exist, sufficient information may be contained in the system documentation.

- For safety software that is in operations or used in analysis or design for several years, the assessment team should consider using an approach similar to the *a posteriori* review described in ANS 10.4. This approach takes advantage of available program development products and program staff, as well as, the experience of the users. The purpose of an *a posteriori* review is to determine if the system produces valid responses. The level of *a posteriori* review may range from a simple demonstration that the software produces reasonable results for a representative sample of inputs or test cases, to a rigorous verification of program requirements, design, coding, test coverage, and evaluation of test results. The team may consider using documented engineering judgments (including their bases) and test results to extrapolate the available existing information to establish functional and performance capabilities.

- Using the *a posteriori* approach for existing software where some documentation does not exist or cannot be found, the assessment may consist primarily of a review of system test procedures, test records, and verification process to ensure the test results are consistent with the software requirements. Documentation of the software requirements is necessary to ensure that future changes to the software are adequately controlled and consistent with system operation as assumed in the facility safety basis.

F.4. ASSESSMENT METHODOLOGY

The assessment planning is to ensure assessments efficiently address the objectives of the assessment. The level of planning will vary depending on scope and complexity of the software system being assessed. The guidance for assessment planning is available in other DOE Guides.[1] In addition, for safety software assessments, the review team should consider the following major activities.

- The team should prepare the assessment plan using the CRAD, and develop a question set with lines of inquiry and detailed attributes, as appropriate, for site-specific applications. The plan should include qualification requirements for team members, a listing of team members and their biographies, a plan for the preassessment visit, and guidance for preparing the report.
- The CRAD is prepared to address safety software, which includes software that performs a safety function as part of an SS and SC system as defined in the facility documented safety analysis (DSA) and technical safety requirements (TSRs). Safety software is an integral part of a safety system. Safety software classification should be consistent with SSC classification unless otherwise justified for case-specific application. The team should use facility-specific DSAs and TSRs for the selection of safety software.
- The team should review DOE O 414.1C, *Quality Assurance*, dated 6-17-05; this Guide; NQA-1-2000; and other applicable standards for assistance in developing the lines of inquiry and to determine their appropriateness for the safety software being assessed. Appendix G of this Guide includes additional industry standards and guidelines.
- The team should use interview methods, as well as, informal discussions with program developers, users, and sponsors to supplement and complement the documented information.

F.5. CRITERIA AND APPROACH

The Criteria and Approach section is divided into the following work activities.

1. Software Project Management and Quality Planning
2. Software Risk Management
3. SCM
4. Software Procurement and Supplier Management
5. Software Requirements Identification and Management
6. Software Design and Implementation

[1] DOE G 414.1-1A, *Management Assessment and Independent Assessment Guide for Use with 10 CFR, Part 830, Subpart A, and DOE O 414.1A, Quality Assurance; DOE P 450.4, Safety Management System Policy; and DOE P 450.5, Line ES&H Oversight Policy*, dated 5-31-01.

7. Software Safety
8. Software Verification and Validation
9. Software Problem Reporting and Corrective Action
10. Training of Personnel in the Design, Development, Use, and Evaluation of Safety Software

Each of these work activities includes the following.

- ***Objective:*** Describes the assessment objective for the work activity and the intended contribution to the adequacy of safety software.
- ***Criteria:*** Suggests characteristics of safety software that should be verified.
- ***Approach:*** Suggests information needed to guide the team in assessing the quality of the safety software. However, the team may choose to select another approach to meet the assessment-specific needs.

Existing QA or other requirements (e.g. procurement) for software may satisfy some of the objectives and criteria for safety software. Previous reviews may also contain information relevant to this assessment that can be cited and used in this assessment. In such situations, this assessment should be limited to objectives and criteria not covered in previous assessments and should not unnecessarily duplicate previous assessments.

A variety of software engineering methods may exist at DOE sites to meet the applicable SQA requirements and work activities. These requirements should be commensurate with the risk associated with a software failure. Factors affecting this risk include the potential impact on safety or operation, complexity of computer program design, degree of standardization, level of customization, state of the art, and comparison with previously proven computer programs.

For each of the ten work activities, the SQA standards and guidance being applied by the contractor should be documented in the assessment report along with the assessment team's judgment of their appropriateness for the specific software application, and the effectiveness of their implementation.

F.5.1 SOFTWARE PROJECT MANAGEMENT AND QUALITY PLANNING

Objective:

Software project management and quality planning should depict the organizational structure that supports the software life-cycle stages and deliverables, and influences and controls the quality of the software.

Criteria:

1. Software project management and quality planning has been implemented depicting organizational structure, responsibilities, and authorities for those managing, performing, and assessing the software projects.

2. SQA activities, software practices, and documentation are periodically assessed.
3. Software quality activities have been effectively implemented.

Approach:

Confirm the existence of project management and QA planning work activity. This may be present in software project management and/or SQA plans that exist either as a stand alone document or embedded in other documents and related procedures. The software project management and software quality planning should identify and/or define the following:

- software project schedule;
- software project scope;
- software engineering activities, including software requirements and design;
- software V&V activities, including reviews and test;
- SCM activities;
- software risk management approach;
- software safety analysis and planning;
- supplier control;
- user and software staff training,
- standards, practices, conventions, and metrics;
- records and document collection, maintenance, and retention; and
- problem reporting and corrective action methods.

Many of the items listed above may be detailed in other documents, for instance software V&V may be detailed in a software V&V plan or in software test plans. It should be noted that this work activity addresses the existence that these items are identified and described. Associated work activities, such as software V&V address the quality of the software V&V work activity being performed as it relates to the grading level.

Determine whether the documents containing the software project management and quality plan are controlled under configuration change control and document control process, and are maintained until the software is retired. This may overlap with the SCM work activity.

Verify that the software project management and quality plan is reviewed and updated, as necessary, for completeness and consistency. This may overlap with the software V&V work activity.

F.5.2 SOFTWARE RISK MANAGEMENT

Objective:

Software risk management is a proactive and disciplined approach to assess and control software risks.

Criteria:

1. Potential software risks are identified as required by the grading level.
2. Likelihood and consequences of the safety software failure are determined.
3. Risks are prioritized.
4. Risk avoidance, mitigation, and/or transfer strategies are created.
5. Risks are monitored.

Approach:

Determine the existence of software risk management planning. This may be evident in a standalone document or embedded in another document. As applicable, ensure that the risk management planning specifies the following:

- scope of the risk management activities;
- risk management policies and process (for both technical and managerial) under which risk management is to be performed are defined;
- identification of the technical and managerial risks and likelihood and potential safety consequences of using software risk taxonomies as a guide;
- establishment of risk thresholds for the safety software application;
- risk avoidance, mitigation, or transfer options; and
- management techniques to address risks throughout project life-cycle, including tracking, decision, and feedback points.

F.5.3 SOFTWARE CONFIGURATION MANAGEMENT

Objective:

Software configuration is defined, maintained, and controlled until the software is retired.

Criteria:

1. Software configuration items are identified, baselined and controlled.
2. A baseline labeling system is established and implemented.
3. In addition, for Level A or Level B custom developed safety software, periodic configuration audits and reviews are conducted and documented.
4. Proposed software changes are documented, evaluated, and approved.
5. Only approved changes are implemented.

Approach:

Review appropriate documents, such as applicable procedures related to safety software change control to determine if an SCM process exists and is effective. This determination is made based on the following actions.

- Verify the existence of documented processes to control, uniquely identify, describe, and document the configuration of each version or update of safety software and its related documentation. This documented evidence may be in either SCM plan or embedded in another software or system level document.
- Verify that a configuration baseline is defined and that it is being adequately controlled. This baseline should include operating system components, any associated runtime libraries, acquired software executables, custom-developed source code files, users' documentation, the appropriate documents containing software requirements, software design, software V&V procedures, test plans and procedures, and any software development and quality planning documents.
- Verify a baseline labeling system has been created that uniquely identifies each configuration item, identifies changes to configuration items by revision, and provides the ability to uniquely identify each configuration.
- Review procedures governing change management for installing new versions of the software components, including new releases of acquired software.
- Review software change packages and work packages to ensure that (1) possible impacts of software modifications are evaluated before changes are made, (2) various software system products are examined for consistency and revised as necessary after changes are made and updated, (3) software is tested according to established standards after changes have been made, (4) changes are evaluated and approved for release by the responsible organization, and (5) software validation is performed as necessary to ensure that the change does not adversely affect the performance of the software.
- Verify by sampling that documentation affected by software changes accurately reflects all safety-related changes that have been made to the software.
- Interview a sample of cognizant line, engineering, and QA managers, and other personnel to verify their understanding of the change control process and commitment to manage changes affecting design, safety basis, and software changes in a formal, disciplined, and auditable manner.
- For custom developed safety software, verify audits or reviews, such as functional configuration audit or physical configuration audit, have been performed.

F.5.4 SOFTWARE PROCUREMENT AND SUPPLIER MANAGEMENT

Objective:

Acquired safety software, either COTS software or custom-developed for DOE, meets the appropriate level of QA based on risk, safety, facility life-cycle, complexity, and project quality requirements.

Criteria:

1. Procurement documents identify the technical and quality requirements.
2. Acquired software meets the technical and quality requirements.
3. Suppliers' QA programs meet or exceed the QA requirements specified in the procurement documents.
4. Procurement documents specify supplier reporting of software defects to the purchaser and the purchaser's reporting of defects to the supplier.

Approach:

Suppliers of acquired software are evaluated to ensure that the safety software is developed under an appropriate QA program and satisfies the specific requirements. The assessment of software procurement process should include the following.

- Determine the existence of safety software technical and QA requirements. These requirements may be embedded in the DOE contractors' or subcontractors' procurement document, software or system design description, or SQA plan. If not documented in the procurement contract, ensure that the supplier has received such technical and QA requirements. This verification may overlap with the Software Requirements Management work activity.
- Verify that the suppliers' QA program has been reviewed and meets or exceeds the procurement specification requirements. The supplier may review the supplier's QA program through supplier assessment, supplier self-declaration, third-party certification, or other similar methods.
- Review evidence that the acquired software was evaluated for the appropriate level of quality. This evidence may be included in the test results, a test summary, supplier site visit reports or supplier QA program assessment reports. This review may overlap with the V&V work activity.
- Review procurement or other documents between the supplier and purchaser for a documented process to report software defects from the supplier to the purchaser and the purchaser to the supplier. This review may overlap with the Problem Reporting and Corrective Action work activity.

F.5.5 SOFTWARE REQUIREMENTS IDENTIFICATION AND MANAGEMENT

Objective:

Safety software functions, requirements, and their bases are defined, documented and managed throughout the safety software life-cycle.

Criteria:

1. The software requirements are documented and consistent with the system safety basis.

2. The functionality, performance, security including user access requirements, interface and safety requirements for the safety software are complete, correct, consistent, clear, testable, and feasible.
3. The documented software requirements are controlled and maintained. Changes to the software requirements are reflected in any and all documentation.
4. Each requirement should be uniquely identified and defined such that it can be objectively verified and validated.

Approach:

Review appropriate safety basis documents, such as DSAs, safety analysis reports, TSRs, procurement specifications and any system documentation to determine if the safety software requirements document is consistent with the safety system design and safety basis. The software requirements may exist either as a standalone document, such as a software requirements specification, or embedded in other system or software level documents.

Determine whether the following types of requirements are addressed as appropriate.

- Verify that the software requirements address functionality, performance, security, safety design inputs, design constraints, installation considerations, operating systems (if applicable), and external interfaces necessary to design the software exist and are documented.
- If access to the system by only authorized users is a requirement, verify that use of software is controlled so that only personnel on authorized user lists apply or maintain safety software.
- Verify that the software requirements are correct, unambiguous, complete, consistent, verifiable, modifiable and traceable as appropriate.
- Verify that acceptance criteria are established in the software requirements for each of the identified requirements. Such criteria should be used for V&V planning and performance as defined in each related life-cycle phase.
- Verify that the software requirement documents are controlled under the configuration change control and document control processes. This may overlap with the SCM activity.
- Verify that software requirement documents are reviewed and updated as necessary. This may overlap with the software V&V work activity.

F.5.6 SOFTWARE DESIGN AND IMPLEMENTATION

Objective:

The safety software design depicting the logical structure, information flow, logical processing steps, data structures and interfaces are defined and documented. The design is properly implemented in the safety software.

Criteria:

1. The design, including interfaces and data structures, is correct, consistent, clearly presented, and feasible.
2. The design is completely and appropriately implemented in the safety software.
3. The design requirements are traceable throughout the software life-cycle.

Approach:

Review the appropriate documents, including design documents, review records, and source code listings. The design may be documented in a standalone document or embedded in other documents.

- The software design description should contain the following information.
  - —A description of the major safety components of the software design as they relate to the software requirements, and any interactions with nonsafety components.
  - —A technical description of the software with respect to control flow, control logic, mathematical model, data structure and integrity, and interface.
  - —A description of inputs and outputs including allowable or prescribed ranges for inputs and outputs.
  - —A description of error handling strategies and the use of interrupt protocols.
  - —The design described in a manner suitable for translating into computer codes.
- Evidence of reviews of the design and code for the appropriate grading exists. This may overlap with the software V&V work activity.
- Evidence of developer testing including any independent testing for the appropriate grading exists.

F.5.7 SOFTWARE SAFETY

Objective

The design of the safety software components are developed in a manner that ensures the software modules will perform their intended safety function in a consistent manner during design bases conditions.

Criteria:

1. Software systems are analyzed at the component level to ensure adequate safeguards are implemented to eliminate or mitigate the potential occurrence of a software defect that could cause a system failure.
2. Safety software is designed with simplicity and isolation of safety functions.
3. Where appropriate fault tolerance and self-diagnostics are implemented in the safety software design.

Approach:

- Review hazard analysis documents to ensure that software component and interface failures are included. This analysis may be part of a software or system level failure modes and effects analysis, fault-tree analysis, event-tree analysis or other similar analysis techniques.
- Review how the identified hazards are resolved. Various methods are used for hazards resolutions, such as eliminations, reduction of exposure, and controlling or minimizing the effects of a hazard.
- Review that the hazard analysis is periodically reassessed throughout the software life-cycle and the changes incorporated as appropriate.
- For Level A safety software, and optionally for Level B safety software, sample safety software modules for proof of design complexity evaluation and isolation of safety functions from nonsafety functions.
- For Level A safety software, and optionally for Level B where safety software modules defects could impact the safe operation of the system, evaluate the software design for the implementation of fault tolerant and/or self-diagnostics techniques.

F.5.8 SOFTWARE VERIFICATION AND VALIDATION

Objective:

The V&V process and related documentation for software are defined and maintained to ensure that (1) the software correctly performs all its intended functions; and that (2) the software does not perform any adverse unintended function.

Criteria:

1. Safety software deliverables have been verified, and validated for correct operation using reviews, inspections, assessments, observation, and testing techniques.
2. Relevant abnormal conditions have been evaluated for mitigating unintended functions through testing, observation, or inspection techniques.
3. Traceability of safety software requirements to software design and acceptance testing has been performed.
4. New versions of the safety software are verified and validated to ensure that the safety software meets the requirements and does not perform any unintended functions.
5. V&V activities are performed by competent staff other than those who developed the item being verified or validated. This may overlap with the training work activity.

Approach:

Review appropriate documents, such as SQA plans, review plans, walkthrough records, peer review records, desk check records, inspection reports, test plans, test cases, test reports, system qualification plans and reports, and supplier qualification reports to determine whether—

- management process exists for performing V&V and management and independent technical reviews;
- reviews and inspections of the software requirement specifications, procurement documents, software design, code modules, test results, training materials, and user documentation have been performed by staff other than those who developed the item;
- software design was performed prior to the safety software being used in operations;
- for design V&V—

  —results of the safety software V&V are documented and controlled;

  —V&V methods include any one or a combination of design reviews, alternate calculations, and tests performed during program development; and

  —the extent of V&V methods chosen are a function of (1) the complexity of the software; (2) the degree of standardization; (3) the similarity with previously proved software; and (4) the importance to safety; and
- for test V&V—

  —documentation for development, factory or acceptance testing, installation, and operations testing exists;

  —documentation includes test guidelines, test procedures, test cases including test data, and expected results;

  —results documentation demonstrates successful completion of all test cases or the resolution of unsuccessful test cases and proves direct traceability between the test results and specified software design;

  —test V&V activities and their relationship with the software life-cycle are defined;

  —software requirements and system requirements are satisfied by the execution of integration, system and acceptance testing;

  —acceptable methods for evaluating the software test case results include (1) analysis without computer assistance, (2) other validated computer programs, (3) experiments and test, (4) standard problems with known solutions, and (5) confirmed published data and correlations;

  —traceability exists from software requirements to design and testing, and if appropriate, to user documentation; and

  —hardware and software configurations pertaining to the test V&V are specified.

F.5.9 SOFTWARE PROBLEM REPORTING AND CORRECTIVE ACTION

Objective:

Formal procedure for software problem reporting, and corrective action for safety software errors and failures are established, maintained, and controlled.

Criteria:

1. Documented practices and procedures for reporting, tracking, and resolving problems or issues are defined and implemented.
2. An evaluation process exists for determining if the reported problem is a safety software defect, error, or something else.
3. Organizational responsibilities for reporting issues, approving changes, and implementing corrective actions are identified and found to be effective.
4. For safety software defects and errors, the defect or error is correlated with the appropriate software engineering elements, identified for potential impact, and all users are notified.
5. For acquired safety software, procurement documents identify the requirements to both the supplier and purchaser to report problems to each other.

Approach:

Review documents and interview facility staff for the problem reporting and notification process to determine whether—

- a formal procedure exists for software problem reporting and corrective action development that addresses software errors, failures, and resolutions;
- problems that impact the operation of the software are promptly reported to affected organizations;
- corrections and changes are evaluated for impact and approved prior to being implemented;
- corrections and changes are verified for correct operation and to ensure that no side effects were introduced;
- preventive measures and corrective actions are provided to affected organizations in a timely manner; and
- the organizations responsible for problem reporting and resolution are clearly defined.

F.5.10 TRAINING PERSONNEL IN THE DESIGN, DEVELOPMENT, USE, AND EVALUATION OF SAFETY SOFTWARE

Objective:

Personnel are trained/qualified and capable of performing assigned work. Continuing training to personnel to maintain job proficiency is provided.

Criteria:

1. A training or indoctrination program exists for each of the following personnel assignments:

   —safety software analysis,

—software development (concept to retirement),

—operations and use, and

—assessment or evaluation of safety software.

2. The training/indoctrination provides for continuing education and training to improve their performance and proficiency.

3. Training/indoctrination is commensurate with the scope, complexity, and importance of the tasks and the education, experience, and proficiency of the person.

Approach:

- Review training records or other documentation and conduct interviews to confirm a training or indoctrination program exists for each of the personnel assignments listed above.
- Verify the training program provides for continuing education.
- Verify the training program is adequate and appropriate for the scope, complexity, and importance of the task being performed.

F.6. REPORT FORMAT

The report is intended for cognizant facility managers and DOE line management, and should include the sections described below. The report should conform to security requirements, undergo classification review if needed, and should not contain classified information or Unclassified Controlled Nuclear Information.

1. **Title Page (Cover).** The cover and title page state the name of the site, facilities assessed, and dates of assessments.

2. **Signature Page.** All team members, signifying their agreement as to the report content and conclusions reached in the areas to which they were assigned, should sign a signature page. In the event all team member signatures cannot be obtained due to logistical considerations, the assessment team leader should obtain members' concurrence and sign for them.

3. **Table of Contents.** The table of contents should identify all sections and subsections of the report, illustrations, charts, and appendices.

4. **Acronyms.** Include a list of acronyms used in the assessment report.

5. **Executive Summary.** The executive summary should provide an overview of the assessment scope, any tailoring, and assessment results.

6. **Introduction.** The introduction should provide information and background regarding the site, facility, system, team composition, methodology, and any definitions applicable to the review.

7. **Tailoring.** Identify any tailoring of the criteria and guidelines provided in this CRAD. State the basis for the tailoring.

8. **Assessment Results.** State whether the assessment criteria are satisfied and describe any exceptions. Summarize opportunities for improvement and include a qualitative conclusion regarding the ability of the system to perform its safety functions in its current condition and to remain reliable over its life-cycle. Recommended actions may also be included. Note any work activities that were not assessed and any limitations to the qualitative conclusion. A detailed discussion of results in each work activity that was assessed should be included as a separate attachment or appendix.

9. **Lessons Learned.** Identify lessons learned that may be applied to future reviews.

10. **Detailed Results.** In each work activity assessed, include sufficient detail to enable a knowledgeable individual to understand the results. The suggested format for this section is as follows.

    - Is the criterion met? [Yes/No]
    - How the review was conducted (include lists of documents reviewed, including any system software documentation and QA, and titles of persons interviewed).
    - System operability issues or concerns.
    - Opportunities for improvement.
    - Recommended changes to criteria and guidance.

11. **Documents and References.** Title, number, revision, and issue dates.

12. **Assessment Data.** Attach assessment records, including lines of inquiry, pertinent assessor notes, and other relevant work papers.

13. **Biographies of Team Members.** Include brief biographies of all assessment team members.

APPENDIX G. REFERENCES

The following referenced documents were used in developing the information contained in this Guide. Some of these documents, such as DOE Orders and the quality assurance Rule, may be obtained through the online DOE Directives, Regulations, and Standards Web site: http://www.directives.doe.gov. Other documents, such as the American Society of Mechanical Engineers(ASME), American Society for Quality, and International Electrotechnical Commission (IEC) standards and guidance documents may be purchased or obtained from the sponsoring organizations.

1. 10 CFR 50, Appendix B, Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants.

2. 10 CFR 63, Disposal of High-Level Radioactive Wastes in a Geologic Repository at Yucca Mountain, Nevada.

3. 10 CFR 830, Nuclear Safety Management.

4. 10 CFR 70, Domestic Licensing of Special Nuclear Material.

5. American National Standards Institute (ANSI)/American Nuclear Society (ANS) 10.4-1987 (R1998), *Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry,* American Nuclear Society, 1998.

6. ANSI/American Society for Quality Control E4-1994, *Specifications and Guidelines for Quality Systems for Environmental Data Collection and Environmental Technology Programs—Requirements with Guidance for Use,* American Nuclear Society, 1994.

7. American Society of Mechanical Engineers (ASME), Re: Comments on the Benefits of National Nuclear Quality Assurance Standards for NNSA and DOE Nuclear Activities and Oversight, Letter to Linton F. Brooks, NNSA (2002).

8. ASME NQA-1-1997, *Quality Assurance Requirements for Nuclear Facility Applications,* American Society of Mechanical Engineers, 1997.

9. ASME NQA-1-2000, *Quality Assurance Program for Nuclear Facilities,* American Society of Mechanical Engineers, 2001.

10. ASME NQA-1A-1999, Addenda to ASME NQA-1-1997, *Quality Assurance Requirements for Nuclear Facility Applications,* American Society of Mechanical Engineers, 1999.

11. CE-1001-STD-Rev.2, *Standard for Software Engineering of Safety Critical Software*, CANDU Computer Systems Engineering Centre for Excellence, Atomic Energy of Canada, Ltd., and Ontario Power Generation, Inc., 1999.

12. Christensen, Mark J., and Richard H. Thayer, *The Project Manager's Guide to Software Engineering's Best Practices,* Institute of Electrical and Electronics Engineers (IEEE) Computer Society Press, 2001.

13. Capability Maturity Model Integration (CMMI) Product Team, Capability Maturity Model Integration, Version 1.1, CMMI for Systems Engineering, Software Engineering, Integrated Process Development, and Supplier Sourcing (CMMI-SE/SW/IPPD/SS, V1.1), Continuous Representation, CMU/SEI-2002-TR-011, ESC-TR-2002-011,Carnegie Mellon University (CMU) Software Engineering Institute (SEI), 2002.

14. CMMI Product Team, Capability Maturity Model Integration, Version 1.1, CMMI for Software Engineering (CMMI-SW, V1.1), Staged Representation, CMU/SEI-2002-TR-029, ESC-TR-2002-029, CMU SEI, 2002.

15. Defense Nuclear Facilities Safety Board (DNFSB) Recommendation 2002-1, Quality Assurance for Safety-Related Software, Defense Nuclear Facilities Safety Board, September 2002.

16. DNFSB/TECH-25, *Quality Assurance for Safety-Related Software at Department of Energy Defense Nuclear Facilities,* Defense Nuclear Facilities Safety Board Technical Report, January 2000.

17. DNFSB/TECH-31 *Engineering Quality into Safety Systems,* Defense Nuclear Facilities Safety Board Technical Report, March 2001.

18. Department of Defense (DoD) Instruction 5000.61, DoD Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A), U.S. Department of Defense, dated 5-13-03.

19. DoD MIL-STD-882D, *Standard Practice for System Safety,* U.S. Department of Defense, dated 2-10-2000.

20. Department of Energy (DOE), Office of Environment, Safety and Health, Designation of Initial Safety Analysis Toolbox Codes, Memorandum to Linton Brooks, Defense Programs and Jessie Hill Roberson, Office of Environmental Management, March 28, 2003.

21. DOE Letter and Report, Quality Assurance for Safety-Related Software at Department of Energy Defense Nuclear Facilities, DOE Response to TECH-25, U.S. Department of Energy, October 2000.

22. DOE G 414.1-1A, *Management Assessment and Independent Assessment Guide for Use with 10 CFR, Part 830, Subpart A, and DOE O 414.1A, Quality Assurance; DOE P 450.4, Safety Management System Policy; and DOE P 450.5, Line ES&H Oversight Policy,* dated 5-31-01.

23. DOE G 414.1-2, *Quality Assurance Management System Guide for use with 10 CFR 830.120 and DOE O 414.1,* dated 6-17-99.

24. DOE G 420.1-1, *Nonreactor Nuclear Safety Design Criteria and Explosives Safety Criteria Guide for use with DOE O 420.1, Facility Safety,* dated 3-28-00.

25. DOE O 414.1B, *Quality Assurance,* dated 4-29-04.

26. DOE O 414.1C, *Quality Assurance,* dated 6-17-05.

27. DOE, *Framework for Grading Safety Software for DOE Directive,* Work Paper, U.S. Department of Energy, dated 4-22-04.

28. DOE-RW-0333P, *Quality Assurance Requirements and Description,* Office of Civilian Radioactive Waste Management Program, dated 8-23-04.

29. DOE-STD-1172-2003, *Safety Software Quality Assurance Functional Area Qualification Standard,* dated 12-03.

30. DOE-STD-3009-94, *Preparation Guide for U.S. Department of Energy Nonreactor Nuclear Facility Documented Safety Analyses,* dated 7-94.

31. Herrmann, Debra S., *Software Safety and Reliability: Techniques, Approaches, and Standards of Key Industrial Sectors,* IEEE Computer Society, 2000.

32. International Atomic Energy Agency (IAEA) Safety Guide (SG) Series No. NS-G-1.1, *Software for Computer Based Systems Important to Safety in Nuclear Power Plants,* IAEA, 2000.

33. IAEA Safety Series No. 50-C/SG-Q, *Quality Assurance for Safety in Nuclear Power Plants and other Nuclear Installations,* Code and Safety Guides Q1–Q14, IAEA, 1996.

34. IAEA Technical Reports Series No. 397, *Quality Assurance for Software Important to Safety,* IAEA, 2000.

35. IEC 61508 Parts 1–7, *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems,* IEC, 1998.

36. IEC 61511 Parts 1–3, *Functional Safety—Safety Instrumented Systems for the Process Industry Sector,* IEC, 2003.

37. IEC 880, *Software for Computers in the Safety Systems of Nuclear Power Stations,* IEC, 1986.

38. IEEE Std 730-2002, *Standard for Software Quality Assurance Plans,* IEEE, 2002.

39. IEEE Std 828-1998, *IEEE Standard for Software Configuration Management Plans,* IEEE, 1998.

40. IEEE Std 1008-1987(R1993), *Software Unit Testing,* IEEE, 1993.

41. IEEE Std 1012-1998, *IEEE Standard for Software Verification and Validation,* IEEE, 1998.

42. IEEE Std 1012a-1998, *IEEE Standard for Software Verification and Validation—Supplement to 1012,* IEEE, 1998.

43. IEEE Std 1028-1997, *IEEE Standard for Software Reviews,* IEEE, 1997.

44. IEEE Std 1042-1987, *IEEE Guide to Software Configuration Management,* Section 3.3.4, "Audits and Reviews," IEEE, 1987.

45. IEEE Std 1058-1998, *IEEE Standard for Software Project Management Plans,* IEEE, 1998.

46. IEEE Std 1063-1987(R1993), *IEEE Standard for Software User Documentation*, IEEE 1993.

47. IEEE Std 1219-1993, *Standard for Software Maintenance,* IEEE, 1993.

48. IEEE Std 1228-1994, *IEEE Standard for Software Safety Plans,* IEEE, 1994.

49. IEEE Std 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology,* IEEE, 1990.

50. IEEE Std 730-2002, *IEEE Standard for Software Quality Assurance Plans,* IEEE, 2002.

51. IEEE Std 7-4.3.2-2003, *IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations,* IEEE, 2003.

52. IEEE Std 829-1998, *IEEE Standard for Software Test Documentation,* IEEE, 1998.

53. IEEE Std 830-1998, *Recommended Practice for Software Requirements Specifications,* IEEE,1998.

54. IEEE Std 982.1-1998, *IEEE Standard Dictionary Of Measures To Produce Reliable Software,* IEEE, 1998.

55. IEEE Std 982.2-1988, *IEEE Guide For The Use Of IEEE Standard Dictionary Of Measures To Produce Reliable Software,* IEEE, 1988.

56. IEEE/EIA Std 12207.0-1996, *Industry Implementation of International Standard ISO/IEC 12207: 1995: Information Technology—Software Life Cycle Processes,* IEEE and the Electronic Industries Alliance (EIA), 1996.

57. IEEE/EIA Std 12207.1-1997, *Industry Implementation of International Standard ISO/IEC 12207: 1995: Information Technology—Software Life Cycle Processes—Life Cycle Data*, IEEE and EIA, 1997.

58. IEEE/EIA Std 12207.2-1997, *Industry Implementation of International Standard ISO/IEC 12207: 1995: Information Technology—Software Life Cycle Processes—Implementation Considerations,* IEEE and EIA, 1998.

59. IP 2000-2, Implementation Plan for Defense Nuclear Facilities Safety Board Recommendation 2000-2, *Configuration Management, Vital Safety Systems,* dated 10-31-00.

60. IP 2002-1, Implementation Plan for Defense Nuclear Facilities Safety Board Recommendation 2002-1, *Quality Assurance for Safety-Related Software at Department of Energy Defense Nuclear Facilities,* dated 3-13-03.

61. International Organization for Standardization (ISO) 9000-3, *ISO Quality Management and Quality Assurance Standards*—Part 3: *Guidelines for the Application of ISO*

9001:1994 to the Development, Supply, Installation and Maintenance of Computer Software, ISO, 1997.

62. ISO 9001-1994, *Quality Systems—Model for Quality Assurance in Design, Development, Production, Installation and Servicing*, ISO, 1994.

63. ISO 9001-2000, *Quality Management Systems—Requirements*, ISO, 2000.

64. ISO/IEEE Standard 16085, *IEEE Standard for Software Engineering: Software Life Cycle Processes, Risk Management*, IEEE, 2004.

65. Leveson, Nancy, *Safeware: System Safety and Computers*, Addison Wesley, 1995.

66. National Aeronautics and Space Administration (NASA) NASA-STD-2201-93, *Software Assurance Standard*, NASA, 1992.

67. NASA-STD-8719.13A, *Software Safety*, NASA, 1997.

68. Pressman, Roger S., *Software Engineering: A Practitioner's Approach*, McGraw Hill, 1992.

69. Society of Automotive Engineers (SAE), SAE JA1003, *Software Reliability Program Implementation Guide*, SAE, 2004.

70. Sparkman, Debra, *Techniques, Processes, and Measures for Software Safety and Reliability*, Lawrence Livermore National Laboratory, UCRL-ID 108725, 1992.

71. SQAS21.01.00-1999 (Reference Document), *Software Risk Management: A Practical Guide*, Department of Energy Quality Managers Software Quality Assurance Subcommittee, February 2000.