DATE: September 9, 2004

FROM: BRIAN COSTLOW, ACTING DIRECTOR
OFFICE OF MANAGEMENT COMMUNICATIONS, ME-43

TO: DIRECTIVES POINTS OF CONTACT

SUBJECT: DRAFT DOE G 414.1-4, *Safety Software Guide for Use with 10 CFR 830, Subpart A, Quality Assurance Requirements, and DOE O 414.1C, Quality Assurance*

This is to notify you that the subject draft Guide has been posted in the "Draft" section of the DOE Directives portal for simultaneous use and coordination. This Guide provides information plus acceptable methods useful for implementing the safety software quality assurance requirements of DOE O 414.1C, *Quality Assurance*.

Guides are not requirements documents and are not to be construed as requirements in any audit or appraisal for compliance with the parent Policy, Order, Notice, or Manual. Since Guides do not contain requirements, their content is at the discretion of the author. Therefore, comments on Guides should not be designated "major" or "suggested"; they should simply be labeled as "comments."

Guides are reviewed through the Directives System, but are not coordinated using RevCom. Instead they are posted on the directives portal at:
http://www.directives.doe.gov/directives/draft.html

Comments on the Guide are due October 12, 2004. The Office of Environment, Safety and Health has requested a 30-day review of the Guide because it is critical in meeting the Secretarial commitments to the Defense Nuclear Facilities Safety Board Recommendation 2002-1 Implementation Plan by the end of this calendar year.

The draft Guide was prepared with input from the Department's Safety Software Subject Matter Expert (SME), in which some SMEs participated on the Guide writing team. DOE organizations are encouraged to involve their Safety Software SME Panel member in commenting on the draft Guide to ensure the benefit of their expertise.

The following procedures should be followed for the submission of comments:

Directives Points of Contact at Headquarters Elements: Submit one set of consolidated comments to the originator of the Guide: Bud Danielson, EH-31, Room 5038, Bldg. 270, Germantown, facsimile: 301-903-4120; or INTERNET address: bud.danielson@eh.doe.gov.

Send an additional copy of comments to LaVerne Fuller, ME-43, Room 4B-172, Forrestal, facsimile: 202-586-1972, or to: laverne.fuller@hq.doe.gov.

Directives Points of Contact at Field Elements: will submit consolidated comments to their appropriate Lead Program Secretarial Office. If appropriate, the package submitted by Field Elements may contain comments provided by contractors.

Contractors will submit comments directly to their appropriate Field Elements.

Questions concerning the content of the Guide should be directed to Bud Danielson at (301) 903-2954. Questions on the directives system should be directed to LaVerne Fuller at (202) 586-1996.

Attachment

G

NOT MEASUREMENT
SENSITIVE

DOE G 414.1-4
DRAFT XX-XX-04

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 to be 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.

AVAILABLE ONLINE AT:
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 (EH), 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
10001 Germantown Road
Germantown, MD 20874-1290
Phone: 301-903-2954
Fax: 301-903-6172
e-mail: bud.danielson@hq.doe.gov

Chip Lagdon
U.S DOE
Director, Quality Assurance Programs
1000 Independence Avenue SW
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.

BACKGROUND

Over the past 10 years or so, with safety software use beginning prior to 1995, the use of digital computers or programmable electronic logic systems has increased significantly in safety applications at nuclear facilities across the Department of Energy (DOE or Department) complex and also in applications associated with various Industry related projects and operations in general. This is especially true in the work scope associated with the Nuclear Regulatory Commission in dealing with commercial nuclear power reactor operations and also within nonnuclear industry processes, such as that tied to chemical productions as associated with petroleum. Over this span of time, DOE and Industry concerns have increasingly developed requiring more direct focus regarding the quality assurance of safety software being used for both human and environmental protection purposes. This includes safety software being used at the Department's nuclear facilities to provide protection for the public, the workers and the environment. Industry, from the early-on applications of digital safety systems, has been taking aggressive, logical action or steps to address safety critical software applications through the development and implementation of standards and new, plus revised, regulatory requirements.

Further, DOE awareness of, and direct experience with, safety software use during the past 10 years has led to increased concerns tied to, such as safety-related decision making, the quality of the software used to design or develop safety-related controls, the proficiency of personnel implementing and using the software, and the performance of various safety-related functions. Typical Department safety software application experience situations over the past years is clearly expressed within the recent example addressed in a General Accounting Office (GAO) report dated April 2004 related to Yucca Mountain, "Persistent Quality Assurance Problems Could Delay Repository Licensing and Operation." In a June 2003 audit, DOE auditors discovered recurring software problems that affected confidence in the adequacy of software codes. Specifically, the auditors found ineffective software processes in five areas: technical reviews, software classification, planning, design, and testing. The auditors found several of the software development problems to be similar to previously identified problems, indicating that previous actions were ineffective in correcting the problems. For example, auditors again noted instances of noncompliance with software procedures. They also concluded that technical reviews during software development were inadequate, even though documentation indicated that corrective actions for this condition had been completed three (3) months before the 2003 audit. Auditors also noted poorly defined roles and responsibilities as a cause of problems identified in the technical review of software, even though DOE had taken actions under its 2002 corrective action plan to clarify roles and responsibilities.

Beginning as early as the year 2000, the Department had actually initiated its own planning programs to address more specifically safety software use concerns at nuclear facilities. It is important to note that the Department also recognized and considered the issues identified by Defense Nuclear Facilities Safety Board (Board) Recommendation 2002-1, Quality Assurance for Safety-Related Software, in light of the required protection of the public, the workers and the environment. The Department, consistent with the Board, continued to agree that potential weaknesses in safety software applications do exist across the Departmental complex which could have an effect on protection required by the implementation of various nuclear facility safety systems. The Department committed itself to develop an implementation plan as stated within the Secretary's letter of November 21, 2002. Subsequently, a software quality assurance implementation plan (SQAIP) was officially developed to address safety software quality assurance issues and was issued on March 13, 2003.

The Department's SQAIP defines the multiple actions and processes that will be taken and implemented respectively to ensure the quality of safety software used at defense nuclear facilities. More specifically as it pertains to this Guide, Section 4.3 of the SQAIP includes a commitment by the Department to make improvements in the directives system through new and revised DOE Policies, Orders, Manuals, Standards, or Guides as determined appropriate. This ultimately resulted in the development and issuance of DOE O 414.1C, Quality Assurance, which includes safety software requirements to complement the requirements of Title 10 Code of Federal Regulations (CFR) 830 Rule, Subpart A, Quality Assurance, regarding nuclear facility safety requirements and specific support guidance in accordance with this Departmental Guide (G), DOE G 414.1-4, Safety Software Guide.

CONTENTS

CONTENTS (continued)

1. INTRODUCTION

1.1 PURPOSE

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

DOE has promulgated the supplemental safety software requirements and this guidance for the purpose of controlling or eliminating the hazards and associated postulated accidents posed by nuclear 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 over-arching QA Program of which safety software is one set of work activities within its scope. This Guide includes software application practices covered by appropriate national and international consensus standards and various acceptable processes currently in use at DOE facilities. This guidance is also considered to be of sufficient rigor and depth to ensure upon its proper use acceptable reliability of safety software at DOE nuclear facilities based on the associated risks and complexity of the operations.

This guidance should be used by organizations to help determine and support the path necessary to address any possible design or functional implementation deficiencies that might exist and to reduce operational hazards-related risks to an acceptable level. In addition, attributes such as the facility lifecycle stage and the hazardous nature of each facility's scope of operations should be considered when making use or application determinations regarding this Guide. It is appropriate to clarify further that alternative methods outside of that described in DOE G 414.1-4 may be used by the responsible organizations provided that the methods implemented result in adequate or satisfactory compliance with the requirements of the 10 CFR 830 and DOE O 414.1C.

2. INTENDED USE AND RESPONSIBILITIES

2.1 SCOPE

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

[1] DOE Guide, G 414.1-2A, Quality Assurance Management System Guide for use with 10 CFR 830 Subpart A and DOE O 414.1B.

The Department's objectives for safety software requirements include:

- Grading SQA requirements based on risk, safety, facility lifecycle, complexity, and project quality requirements;
- Applying SQA requirements to software lifecycle phases;
- Developing procurement controls for acquisition of computer software and hardware that are provided with vendor-developed software and/or firmware;
- Documenting and tracking customer requirements;
- Managing software configuration throughout the lifecycle phases;
- Performing verification and validation[2] processes;
- Performing reviews of software configuration items, including safety implications that will address considerations, such as failure analysis and fault tolerance; and
- Training of personnel who use and apply software in safety applications

The scope of this Guide is bounded by the safety software definitions are stated in the QA Order. Safety software includes both safety system software and safety analysis and design software.

Safety system software, is software which performs a safety system function as part of a structure, system, or component (SSC) that has been functionally classified as safety class (SC) or safety significant (SS) as per 10 CFR 830.2. Safety system software includes human-machine interface software, network interface software, and programmable logic controller (PLC) programming language software. Safety system software also includes safety management databases that are not part of an SSC but whose operation or malfunction can directly affect SS and SC SSC function.

Safety analysis and design software, is software which_is not part of an SSC but is used in the safety classification, design, and analysis of nuclear facilities to ensure the proper accident analysis of nuclear facilities; the proper analysis and design of safety SSCs; and the proper identification, maintenance, and operation of safety SSCs.

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

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

2.2 SAFETY SOFTWARE APPLICATION TYPES

Within the definitions of safety system software and safety analysis and design software, five basic software applications can be identified. Safety system software is composed of three safety

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

applications: (1) instrumentation and control (I&C) process monitoring and control applications; (2) networking and interface applications; and (3) safety management and administrative applications. Safety analysis and design software identifies the remaining two application types: (1) safety analysis applications and (2) design and analysis applications. These five types are further described and examples are included in the following paragraphs.

Instrumentation and control applications are those applications where software and firmware provide control and monitoring functionality for such components as valves and switches, including all programmable logic controllers (PLC), supervisory and control data acquisition systems (SCADAs), distributed control systems (DCS) and hybrid systems consisting of a mix of DCS and PLC/Human Machine Interface (HMI) features.

Networking and interface applications include those software applications that are used for communications with or interface with SSCs that perform safety functions.[3] These include networking protocols and security functions for Local Area Networks (LAN) and associated security profiles.

Safety management and administrative control applications include database applications used in the safety management and administrative controls associated with safety systems.[4] These applications are included in a facility safety basis as actions to be performed to prevent a safety basis violation from occurring. Examples of such systems include software used for inventory and material tracking waste drum or container hazard assessment calculations, and process simulation applications for safety systems operations training.

Safety analysis applications are used for consequence analysis of potential accidents and the evaluation of design basis events. Examples include criticality, fire, chemical and radiological dispersion, and leak path factor application software used for safety basis analyses.

Lastly, safety software applications include software used for the design and analysis of safety structures, systems, and components for the facility. These applications include software used in structural; electrical; mechanical; heating, ventilation and air conditioning (HVAC); criticality safety; fire protection design and analysis decisions.

2.3 SOFTWARE SOURCE TYPES

Software typically can be considered either custom developed or acquired. Further defining these two basic types identifies specific characteristics and attributes that can be used to select the applicable practices and approaches for performing safety software quality work activities. The software source types are one dimension to determine how the safety software quality assurance work practices are applied. Five types of software are commonly used in DOE applications: (1) custom developed, (2) configurable, (3) acquired, (4) utility calculations, and (5) commercial design and analysis tools.

[3] Page 3, *Framework for Grading Safety Software for DOE Directive Work Paper*, April 22, 2004.

[4] Op. cit., page 1.

Developed and acquired software types as discussed in American Society of Mechanical Engineers (ASME) NQA-1 are compatible with these five source types. Developed software as described in ASME NQA-1 is directly associated with custom developed, configurable and utility calculations. Acquired software included in this Guide is easily mapped to that of acquired software in ASME NQA-1. ASME NQA-1 uses acquired and procured software terms interchangeably.[5] This Guide includes an additional software source type of commercial analysis and design software that is not directly related to either developed or acquired. DOE is limited to the same quality assurance controls on commercial analysis and design software as with commercial-off-the-shelf (COTS) software. Safety software quality requirements can only be specified through work activities associated described in contractual agreements with the supplier of the facility design and analysis services.

Custom developed software is that software that is built specifically for a DOE application, or to support the same function for a related government organization. It may be developed by DOE, one of its M&O contractors, or contracted with a software company through the procurement process. Examples of custom developed type of software could include material inventory and tracking database applications, accident consequence applications, and control system applications.

Configurable software is a commercially available software or firmware that allows user to modify the structure and functioning of the software in a limited way to suit user's needs. An example of this will be the software associated with programmable logic controllers.

Acquired software is generally supplied through basic procurements, two-party agreements, or other contractual arrangements. Acquired software includes COTS software, such as operating systems, database management systems, compilers, software development tools (e.g., Power Tools for Windows-PTW), as well as, the commercial calculational software and spreadsheet tools (e.g., Mathsoft's MathCad and Microsoft's Excel, respectively). Firmware is acquired software since the computer program cannot be changed after it has been downloaded into the computer hardware.

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 software quality assurance work activities as custom developed software.[6] With utility calculation software, it is important to recognize the difference between quality assurance of the algorithms, macros, and logic that performs the calculations versus quality assurance of the COTS spreadsheet software itself. Utility calculation software includes the associated data sets, configuration information and test cases for validation and/or calibration.

[5] Page 105, 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.

[6] Page 227, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications,* Part 4 *Subpart 4.1 Section 101.1*, American Society of Mechanical Engineers, New York, New York, 2001.

Design and analysis software can be proprietary, freeware or available for purchase. Proprietary software is typically custom developed software generally not available to the public, but used by the owner as part of a service. An example would be where DOE or its M&O contractor contracts for design services. The design service provider (aka the supplier) uses their independently developed software (without DOE involvement or support). DOE then receives a completed design. Purchased software is one which is available publicly and is generally procured directly from a supplier. Procurement contracts can be enhanced to require details of the quality assurance work activities performed on the software product. DOE or its contractor in performing design and analysis activities then uses this software. Examples include ANSYS, and ABACUS.

2.4 GRADED APPLICATION

Proper application of the Order can be enhanced by grading safety software requirements. Safety software grading levels are described in terms of safety consequence and regulatory compliance. The safety software grading levels are the second dimension to determine how the safety software quality assurance work practices are applied Grading levels of all safety software are defined as:

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

1. Failure could have an adverse effect on nuclear safety systems (i.e., Safety Class or Safety Significant SSCs), toxic material or chemical hazard protection systems that are credited in the facility safety analysis for protecting against or limiting exposure to the general public and workers below regulatory or evaluation guidelines.
2. Failure could result in non-conservative safety analysis, misclassification of facilities and SSCs, incorrect monitoring and recording of radiological exposures to workers and the public, or inappropriate safety related decisions.

Level B: This grading level includes low safety consequence software applications that meet one or more of the following criteria and does not meet the Level A criteria:

1. Failure that would cause a reduction in the degree of safety or defense-in-depth.
2. Failure that would impact safety management decisions regarding a facility or system operating activity (e.g., software whose failure would not impact performance of a safety function, but could result in: missed surveillances; confusion regarding system status; or, noncompliance with nuclear safety regulatory laws, environmental permits or regulations and/or commitments to compliance).

Using the grading levels and the safety software source types above, select and apply the following SQA work activities in accordance with ASME NQA-1 and supplemented by national or international consensus standards. This Guide provides acceptable implementation strategies for these practices as identified here.

1. Software project management
2. Software risk management
3. Software configuration management
4. Procurement and vendor management
5. Software requirements identification and management
6. Software design and implementation
7. Software safety design
8. Verification and validation
9. Problem reporting and corrective action
10. Training of personnel in the design, development, use and evaluation of safety software

The determination of what constitutes safety software is made by the organizations using the software based upon the requirements in DOE O 414.1C, *Quality Assurance,* and 10 CFR 830 Subparts A and B. The use of the software determines whether it is safety related. Therefore, the organization applying the software is responsible to identify, evaluate and designate the software as safety software and then ensure that the software development and operations have followed the appropriate Quality Assurance procedures.

2.5 RESPONSIBILITY FOR SAFETY SOFTWARE

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

2.6 SAFETY SOFTWARE QUALITY PROGRAM

The scope of the Department's Quality Assurance 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 safety software (items or services). The contractor QA Program includes safety software within its scope. The Order is used to establish the safety software QA requirements to be implemented under the Rule. The contractor should perform safety software work in accordance the quality assurance criteria established in 10 CFR 830 Subpart A, DOE O 414.1C and described within the contractor's QAP. The various chapters of this Guide will discuss application of the QA Criteria to safety software work activities. Table 1 provides an illustration of the QA criteria for 10 CFR 830 applied to the safety software quality assurance work activities.

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

	Project Management	Risk Management	Configuration Management	Procurement and Vendor Management	Requirements Identification and Management	Design and Implementation	Safety Design	Verification and Validation	Problem Reporting & Corrective Action	Training In Design, Development, Use and Evaluation
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 processes documents. For example, an independent assessment may be performed on any safety software quality element.

2.7 SOFTWARE QUALITY ASSURANCE PROGRAM

It is important to note that the Software Quality Assurance (SQA) Program is part of an overall Quality Assurance Program required for nuclear facility operations in accordance with 10 CFR 830 and DOE O 414.1C quality assurance requirements. Regardless of the application or software source type for the safety software, an appropriate level of quality infrastructure should be established and a commitment made to maintain this infrastructure.

A keystone to this infrastructure is the establishment of a SQA Program for safety software. A SQA program establishes the appropriate safety software life cycle practices, including safety design concepts, to ensure that software functions reliably and correctly to perform 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 a SQA program is driven by the intended use of the safety software. More importantly, the rigor of a SQA program should address the risk of use of such software. An effective safety software quality program is one method for avoiding, minimizing or mitigating the risk associated with the use of the safety software.

The goal of a software quality program for safety system software is to apply the appropriate quality practices to mitigate the risk of failure of safety systems caused by failure of the software. A safety system software quality program should apply necessary and sufficient practices to reduce the risk of safety software failure (not system failure) to acceptable and manageable levels. The SQA program 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.

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 also addressed. All three attributes are inter-dependent since compromising the security access could obviously present a potential safety hazard also. If the integrity of either the data or application itself has been compromised either accidentally or maliciously, the safety again could be compromised. So when consideration of safety software is being addressed, the integrity and security issues should likewise be addressed.

Other issues impacting safety software are the availability of trained and knowledgeable personnel to develop and maintain the software, human factor issues, such as understandability of the displays or ambient lighting conditions if interactions with an operator are required, potential EMI/RFI, fault tolerance and common cause failure issues, the fault mode when an exception handler is being used, performance requirements, and proper identification and analysis of functional requirements that have safety, security or integrity implications.

From the foregoing, it can be seen that there are several interdependencies and tradeoffs that should be addressed when integrating software into a 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 verification and validation activities exist are not sufficient by itself to produce safe and reliable software.[7] The lifecycle process should focus upon the safety issues in addition to the basic software quality engineering principles. Both of these concepts are detailed in later sections 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, all 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 the 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 the key to maintaining the integrity of the safety software. Likewise, if human factor issues, such as ambient lighting conditions or user interface ease of use or understandability are important for operational use of the safety software system, the risks need to be addressed either via design or training. For programmable logic controllers or network safety software applications, electromagnetic interference could offer potential risks for operation of the safety software system.

Once this perspective is achieved, then the appropriate software life cycle and system life cycle practices can be identified to minimize the risk of the use of software within a system. Rigor can then be applied commensurate with the risk associated with a software failure causing a system failure. 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 no true understanding of the issues involved regarding the safety software system. Obviously, there are time and resource constraints that should balance the probability of occurrence and the potential consequences versus an occurrence of the worst case scenario. If the safety systems staff zealously and religiously invokes the strictest rigor for a safety software application for a Level B application, then the application has the potential to never 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

[7] Page 395, *Safeware*, Nancy Leveson, Addison Wesley, 1995.

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

The development and maintenance of a collection, or "toolbox," of multiple-site use, standard solution, SQA-compliant safety software is one of the improvement actions identified by the DOE for safety software. Ultimately, the DOE Safety Software Central Registry (website: http://www.eh.doe.gov/sqa/central_registry.htm) will contain information on a set of quality-assured, configuration-controlled, safety analysis software applications, recognized for DOE-broad, safety basis applications. Six widely applied safety analysis computer codes were originally designated for toolbox consideration, including:

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

The current designated toolbox software and any software recognized in the future as meeting the toolbox equivalency criteria are no different from other custom safety software as defined in Section 2.3. 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 the software noted above may be added to the registry while the older versions are removed. Over time, some of the software may be retired and no longer advised for use in DOE safety analysis. Still others may be added through a formal toolbox-equivalent process, having been recognized as meeting the equivalency criteria. Thus, the Central Registry collection of safety-related 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, as well as, 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 (website: 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 quality assurance 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 the SQA Program was not updated as the SQA standards were revised.

Existing safety software should be identified and controlled prior to evaluation against the graded approach framework in this Guide. The evaluation performed and documented should be adequate to address the correct operation of the safety software in the environment it is being used. This evaluation should include identification of the capabilities and limitations for intended use, any test plans and test cases required to demonstrate those capabilities, and 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

Tom DeMarco aptly stated "You can't control what you can't measure".[10] This truism especially applies to safety software systems. Metrics used throughout the lifecycle should bolster the confidence that the software applications will achieve their mission in a safe and reliable manner. If design measures, testing measures 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 Assurance 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 can be implemented to determine the root causes and eliminate the events that could lead to a reoccurrence. Metrics further provide an indication (qualitative or quantitative) of the improvements or lack there of when a process or work activity has been modified. Metrics are the evidence that an improvement has occurred. Both IEEE 982.1 and 982.2 provide recommendations for what metrics to use and when in the software life cycle phase to the metric is most appropriate.

3.5 USE OF NATIONAL/INTERNATIONAL STANDARDS

The DOE QA Rule and Order require the use of standards to develop and implement a QA Program. National/international standards facilitate a common software quality engineering approach to developing or documenting software based upon a consensus of experts in the

[8] Page 105, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part II Subpart 2.7 Quality Assurance Requirements for Computer Software for Nuclear Facility Applications,* Section 302, American Society of Mechanical Engineers, New York, New York, 2001.

[9] Page 29-32, ANSI/ANS 10.4 - 1987 (R1998), *Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry, Section 11, V&V for Existing Programs*, American Nuclear Society, 1998.

[10] Page 3, "Controlling Software Projects", Tom DeMarco, Yourdon Press, 1982.

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, the ASME is the nationally accredited body for the development of nuclear facility quality assurance standards. The DOE QA Order cites ASME NQA-1-2000 as the appropriate standard QA Programs applied to nuclear-related activities (e.g., safety software). The ten quality assurance criteria of the Rule and the QA Order are mapped to ASME NQA-1-2000 in Appendix C. The Order also requires that additional standards be used to address specific work activities conducted under the QAP, such as safety software. Use of ASME NQA-1 supplemented by other software standards will enable compliance with DOE Order 414.1C requirements.

In the case of ASME NQA-1-2000,[11] Part I, 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) have safety software applicability. In addition, ASME NQA-1-2000 Part II, Subpart 2.7 and Part IV, Subpart 4.1, specifically addresses "*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 this subpart, this standard "provides requirements for the acquisition, development, operation, maintenance and retirement of software." Table 2 provides a cross reference of ASME NQA-1 with the ten SQA work activities in the Order. Although ASME NQA-1-2000 standard provides excellent process guidance for a software quality engineering process for managing a software development, maintenance process, procurement or otherwise acquiring software, the details for safety software guidance are not provided within this standard.

Other national and international standards useful for the safety software analyst are discussed in Appendix D. It should be noted that the use of the standards discussed should promote a robust safety software quality engineering process and a resulting software product that is adequate for safety all the software applications.

Other standards, such as IEEE Std. 7-4.3.2-2003[12] specify 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. Appendix D of this Guide includes references to this and other standards useful in achieving compliance with the DOE QA Order and Rule for safety software work activities.

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

[12] 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, Piscataway, NJ, 2003.

Table 2. ASME NQA-1 Cross Reference to DOE Safety Software Requirements

	Project & Quality Management	Risk Management	Configuration Management	Procurement & Vendor Management	Requirements Identification & Management	Design & Implementation	Safety Design	Verification & Validation	Problem Reporting & Corrective Action	Training In Design, Development, Use & Evaluation
Organization	Req. 1, 1A-1, *200*									
Quality Assurance Program	2A-2, *302* 1A-1, *200*	2A-2, *301*			Req. 1 Req. 2, *100* Sp 4.2, *300*		Req. 2 Sp 4.2, *300*			Req. 2
Design Control		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*	Sp 2.7, *400* Sp 4.1, *400*	Req. 3, *800* Sp 2.7, *400* Sp 4.1, *400*	Sp 2.7, *202* Sp 4.1, *100*	Req. 3, *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, *404*	
Procurement Document Control					Req. 4					
Instructions, Procedures, and Drawings					Req. 5	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*		Req. 6 Sp 2.7, *201* Sp 4.1, *201*		

Table 2. (continued)

	Project & Quality Management	Risk Management	Configuration Management	Procurement & Vendor Management	Requirements Identification & Management	Design & Implementation	Safety Design	Verification & Validation	Problem Reporting & Corrective Action	Training In Design, Development, Use & Evaluation
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										
Test Control								Req. 3, *801.4, 801.5* Req. 11, *400* Sp 2.7, *402.1, 404* Sp 4.1, *402.1, 404*		
Control of Measuring and Test Equipment										
Handling, Storage, and Shipping										

Table 2. (continued)

	Project & Quality Management	Risk Management	Configuration Management	Procurement & Vendor Management	Requirements Identification & Management	Design & Implementation	Safety Design	Verification & Validation	Problem Reporting & Corrective Action	Training In Design, Development, Use & Evaluation
Inspection, Test and Operating Status										
Control of Nonconforming Items									Req. 15 Sp 2.7, *204* Sp 4.1, *404*	
Corrective Action									Req. 16 Sp 2.7, *204* Sp 4.1, *404*	
Quality Assurance Records										
Audits										

ANSI/ANS-10.4-1987[13] 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.

The Canadian standard[14] 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 1 applications.[15] This standard recommends particular detailed outputs for the software life cycle processes, but is not prescriptive in how the outputs should be obtained.

Similarly, the IAEA Standard IEC 880[16] is applicable to Level 1 highly reliable safety systems of nuclear power plants. Like its Canadian counterpart, the IAEA standard advises various approaches to maximize the reliability of the safety systems within a nuclear power plant.

The UK standard "SEMSPLC Guidelines: Safety-Related Application Software for Programmable Logic Controllers" targets programmable logic controllers (PLC) in all industry sectors, including military, nuclear, railway and off-shore oil.

MIL-STD-882D[17] 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"[18] provides general guidance for a software safety effort.

In summary, use of the standards should promote a robust safety software quality engineering process and a resulting software product that is adequate for safety software applications.

4. RECOMMENDED PROCESS

Recognizing that safety software applications within DOE comprise five application types listed in Section 2.3 above, 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

[13] ANSI/ANS-10.4-1987, *guidelines for the verification and validation of scientific and engineering computer programs for the nuclear industry,* American Nuclear Society, La Grange Park, Illinois, reaffirmed 1998.

[14] CE-1001-STD, Revision 2, *Standard for Software Engineering of Safety Critical Software,* CANDU Computer Systems Engineering Centre of Excellence, Atomic Energy of Canada Limited and Ontario Power Generation, Inc., December, 1999.

[15] Op. cit., footnote 6.

[16] IEC 880, *Software for computers in the safety systems of nuclear power stations,* International Electrotechnical Commission, Geneva, Switzerland, 1986.

[17] MIL-STD-882D, *Standard Practice for System Safety,* Department of Defense, 10 February, 2000.

[18] NASA-STD-8719.13A, *Software Safety,* National Aeronautics and Space Administration, September 15, 1997.

should support the planning, 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 software is a subcomponent, utilizes two primary approaches: (1) applying standard 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 used in the design.[19]

Applying standard software engineering and software quality engineering practices are 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 correctly. DOE O 414.1C requires SQA practices, referred to as work activities, for safety software to be performed. Many national and international consensus standards, such as ASME NQA-1, American Nuclear Society 10.4, and the IEEE software engineering series provide detailed guidance for performing the work activities. Section 3.6 of this Guide describes some of these standards.

Software process capability models, such as the Software Engineering Institute's Software Capability Maturity Model (swCMM) and the more integrated model, Capability Maturity Model Integrated (CMMI) are proven tools to assist in the selection of practices to perform for achieving a level of assurance the processes performed will produce the desired level of quality for safety software.

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 lifecycle 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 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 provides the basis for planning, implementing, maintaining, and operating safety software. The work activities for safety software include tasks,

[19] Page 398, *Safeware*, Nancy Leveson, Addison Wesley, 1995.

such as software project planning, software configuration management, 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 safety software source type. Table 3 provides a summary of the mapping between safety software source type, the grading levels, and the 10 work activities. Not all work activities will be applicable for a particular instance of safety software. The Guide indicates where these practices 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. For safety software, project management and quality planning start 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 planning activities and documents

These tasks may be documented in a software project management plan (SPMP), a software quality assurance plan (SQAP), a software development plan (SDP) or similar documents. They also may be embedded in 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 SPMP are initiated early in the project lifecycle and are maintained throughout the life of the project.

The software project management and quality planning should include identifying all tasks associated with the software development and procurement, including procurement of service, 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. Several consensus standards[20,21] 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 the relative emphasis that should be placed on each phase of 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 the 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 software configuration management, risk management, problem

[20] IEEE Std 1058, *IEEE Standard for Software Project Management Plans*, Institute of Electrical and Electronic Engineers, Inc., 1998.

[21] IEEE Std 730, *IEEE Standard for Software Quality Assurance Plans*, , Institute of Electrical and Electronic Engineers, Inc., 2002.

reporting and corrective actions, and verification and validation, 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 applies to custom and configurable software sources types for both Level A and Level B safety software. For Level A and Level B acquired and utility calculation source types, safety software project planning and quality management tasks should be identified and tracked. Where instances of the safety software 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.

5.2.2 Software Risk Management

Software risk management provides a disciplined environment for proactive decision-making to assess continuously what can go wrong, determine what risks are important to deal with, and implement actions to deal with those risks.[22] Because risk management is such a fundamental tool for project management, it is an integral part of software project management. 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 then priorities 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. Without an understanding of the issues associated with safety software applications, regardless of the type of DOE application, risk identification and development of the risk mitigation strategies are unproductive. Identification, tracking, and management of the risks throughout all phases of the project's life cycle, with special emphasis upon tracking the risks associated with costs, resources, schedules, and technical aspects of the project, is vital. Several risk identification techniques are described and detailed in standards and literature.[23,24] 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 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,

[22] *Software Risk Management: A Practical Guide*, Department of Energy Quality Managers Software Quality Assurance Subcommittee, Reference Document SQAS21.01.00 – 1999, February 2000.

[23] Page 417-447, *The Project Manager's Guide to Software Engineering' Best Practices***,** Mark J. Christensen and Richard H. Thayer, IEEE Computer Society Press, 2001.

[24] Appendix C4.6, *Surface Vehicle/Aerospace Recommended Practice-Software Reliability Program Implementation Guide, Risk Management*, Society of Automotive Engineers, SAE JA1003, January, 2004.

- a design that is based upon unrealistic or optimistic assumptions,
- design changes during coding,
- complete and defined 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,
- potential quality concerns with subcontractors or vendors.

The above bulleted list identifies a few potential risks associated with safety software applications. The risks associated with the safety software applications need to be understood and documented. Each risk should be evaluated against its risk thresholds. Different techniques may be used to evaluate the risks, such as 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 and leeway regarding risk management based upon the risk categorization of the safety software application may need to be applied. For a Level A application, all the known risks, 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 a Level B application, the granularity for the risks to be identified, analyzed, prioritized and resolved to an acceptable level of risk and tracked should be determined by the safety system staff.

Further guidance regarding risk management is provided by IEEE Standard 16085-2004,[25] which provides guidance regarding the risk management of acquired, developed, operational, or maintained systems to support the existing organizational risk management processes. The 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

Software Configuration Management (SCM) activities identifies all functions and tasks required to manage the configuration of the software system that includes software engineering items establishing the configuration baselines to be controlled, and software configuration change

[25] ISO/IEEE Std 16085, *IEEE Standard for Software Engineering: Software Life Cycle Processes, Risk Management*, Institute of Electrical and Electronic Engineers, Inc., 2004.

control process.[26] The four areas of software configuration management,[27] (1) configuration identification, (2) configuration control, (3) configuration status accounting, and (4) configuration audits and reviews should be addressed in performing activities associated with software configuration management. This Guide extends ASME NQA-1-2000 software configuration management[28] by including configuration audits and reviews.[29]

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

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 safety software development and operation.

Proposed changes to safety software should be documented, evaluated, and approved for release. Only approved changes should be made to safety 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 safety 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 documents, and that the safety software including all documentation that is being delivered, is complete. Physical configuration audit and functional configuration audit are examples of audits or reviews that should be performed.[30]

All four areas of software configuration management noted above apply to both Level A and Level B safety software for custom developed safety software. This work activity may be graded for both Level A and Level B for all other types of safety software. Grading for this work activity includes the optional performance of configuration audits and reviews for both Level A and Level B configurable, acquired, and utility calculation safety software. Software configuration management work activities can only be applied beginning at the point of control

[26] Page 105, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part II Subpart 2.7 Quality Assurance Requirements for Computer Software for Nuclear Facility Applications,* Section 203, American Society of Mechanical Engineers, New York, New York 2001.

[27] IEEE Std 828-1998*, IEEE Standard for Software Configuration Management Plans, Section 4.3,* Institute of Electrical and Electronic Engineers, Inc. 1998.

[28] Page 16, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part I Requirement 3 Design Control,* Section 802, American Society of Mechanical Engineers, New York, New York, 2001.

[29] IEEE 7-4.3.2 *IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations, Section 5.3.5*, Institute of Electrical and Electronic Engineers, Inc., 2003.

[30] IEEE 1042-1987, *IEEE Guide to Software Configuration Management, Section 3.3.4 Audits and Reviews*, Institute of Electrical and Electronic Engineers, Inc., 1987.

of the safety software. Thus, configuration identification, configuration control and configuration status accounting of acquired safety software can only be applied to the configuration items that have been received from the supplier.

5.2.4 Procurement and Supplier Management

All safety software applications have involvement with procuring software and interactions with suppliers. The procurements may be as basic as the purchase of compilers or other development tools for custom software or as complicated as procuring a complete safety system software control system. Thus, there needs to be a variety of approaches for software procurements and supplier management based upon the level of control the DOE, or its contractors, has on the quality of the software being procured, and the grading level of the safety software.

The procurement contracts and other procurement documentation should include the technical[31] and quality[32] requirements for the safety software. These requirements should be verified for completeness and to assess the quality of the safety 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, verifying the supplier has obtained a certification of the safety software quality from a 3rd party (e.g., SEI, ISO, UL), and accepting the safety software based upon key characteristics (e.g., large user base). Descriptions of these approaches can be found in Appendix G.

Assessing the quality of the safety software is only the one aspect to management of procured software. Requirements for supplier notification of defects, new releases, or other issues[33] that impact the operation of the procured safety software should be documented and agreed upon. Mechanisms for the users of the safety software to report defects and request assistance in operating the safety software also need to be documented and agreed upon.

When only services are being procured from a supplier, and if the software used by the supplier meets the criteria of Level A or Level B safety software, the technical and quality requirements for that software should be specified in the contractual agreements with the supplier. As with purchasing safety software, the service supplier's quality program should be reviewed. Procedures for the supplier to report any defects found in software, or process steps used in providing the service, should be identified and documented.

This work activity has no grading associated with its performance. Both Level A and Level B and all software source types of safety software should fully meet this requirement.

[31] Page 18, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications*, *Part I Requirement 4 Procurement Document Control,* Section 202, American Society of Mechanical Engineers, New York, New York, 2001.

[32] Page 18, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part I Requirement 4 Procurement Document Control,* Section 100*,* Section 802, American Society of Mechanical Engineers, New York, New York, 2001.

[33] Page 105, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part II Subpart 2.7 Quality Assurance Requirements for Computer Software for Nuclear Facility Applications,* Section 301, American Society of Mechanical Engineers, New York, New York, 2001.

5.2.5 Software Requirements Identification and Management

Safety system requirements provide the foundation for the requirements to be implemented in the safety software. These system requirements should be decomposed into requirements specific for the software. The identified software requirements may be documented in system level requirements documents, software requirements specifications, procurement contracts or other acquired software agreements. These requirements should identify functional, performance, security, interface and safety requirements, as well as, installation considerations and design constraints. The requirements should be complete, correct, consistent, clear, testable, and feasible.[34]

Once the safety 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 safety software placed into operations. Safety software requirements should be traceable throughout the software life cycle.[35]

This work activity has no grading associated with its performance. Software requirements identification, management and traceability apply to both Level A and Level B safety software and should fully meet this requirement. However, the detail and format of the safety software requirements may vary with the software source type. Custom developed software most likely will contain a larger number of software requirements than configurable, acquired, utility calculational or commercial design and analysis tool software, and thus, a separate more formal document may be applicable. As indicated in the *Procurement and Supplier Management* work activity, software requirements for acquired software may be documented in any procurement agreements.

5.2.6 Software Design and Implementation

During software design and implementation the safety software is developed, documented, reviewed and controlled. The software design requirements should identify the operating system, function, interfaces, performance requirements, installation considerations, design inputs, and design constraints. The safety software design should be complete and sufficient to meet the software requirements.[36] The design activities and documentation should be adequate to fully describe how the safety software will interface with other system components,[37] and how the software will function internally. Data structure requirements and layouts may be necessary to fully understand the internal operations of the software.

[34] IEEE 830-1998, *IEEE Recommended Practice for a Software Requirements Specification, Section 4,3*, Institute of Electrical and Electronic Engineers, Inc., 1998.

[35] Page 106, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part II Subpart 2.7 Quality Assurance Requirements for Computer Software for Nuclear Facility Applications,* Section 401, American Society of Mechanical Engineers, New York, New York, 2001.

[36] Page 16, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part I Introduction, Section 801.2 Design Control,* American Society of Mechanical Engineers, New York, New York, 2001.

[37] Page 106, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part II Subpart 2.7 Quality Assurance Requirements for Computer Software for Nuclear Facility Applications,* Section 402, American Society of Mechanical Engineers, New York, New York, 2001.

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 is needed for configurable or utility calculations. Whereas for custom software, complete functional and logical designs of the safety software components, the input and output data, and pseudo code may be required to fully understand the safety software design. The safety software design documentation may be combined with the documentation of the software requirements or software source code.[38]

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 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 Fagan inspections can be used to identify defects in source code, as well as, design descriptions and other software development process outputs.

Developer testing is a dynamic technique for detecting software failures prior to system level verification and validation techniques, including acceptance testing. Developer testing can be very structured and formal, use automated tools or be less formal. In addition to functional testing, structural, timing (performance testing), stress, security, and human-factors testing should be performed. Various techniques,[39,40] such as error seeding, equivalence class testing, branch and path testing, statistical-based and boundary value testing can be used to improve the efficiency of testing.

The software design and implementation work activity for Level A and Level B custom developed safety software should fully meet this requirement. For this software source 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 are 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 safety software for Level A and Level B may be graded for this work activity. This grading should include fully performing the design work activities as with custom developed safety 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

[38] Page 16, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part I Introduction, Section 801.2 Design Control,* American Society of Mechanical Engineers, New York, New York, 2001.

[39] Page 595-629, *Software Engineering A Practitioner's Approach*, Roger Pressman, McGraw Hill, 1992.

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

performance testing. This work activity does not apply to acquired or commercial design and analysis safety software source types.

Table 3. Mapping Safety Software Source Types and Grading Levels to Work Activities

SQA Work Activity	Level A					Level B				
	Custom	Configurable	Acquired	Utility Calcs	Commercial D & A	Custom	Configurable	Acquired	Utility Calcs	Commercial D & A
Software project mgmt & quality planning	Full	Full	Grade	Grade	n/a	Full	Full	Grade	Grade	n/a
Software risk mgmt	Full	Full	Full	Full	n/a	Grade	Grade	Grade	Grade	n/a
Software configuration mgmt	Full	Grade	Grade	Grade	n/a	Full	Grade	Grade	Grade	n/a
Procurement & vendor mgmt	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full
Software requirements identification & mgmt	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full
Software design & implementation	Full	Grade	n/a	Grade	n/a	Full	Grade	n/a	Grade	n/a
Software safety design	Full	Full	Full	n/a	n/a	Grade	Grade	Grade	n/a	n/a
V&V	Full	Full	Full	Grade	n/a	Grade	Grade	Grade	Grade	n/a
Problem reporting & corrective action	Full	Full	Full	Grade	Full	Full	Full	Full	Grade	Full
Training	Full	Full	Full	Full	n/a	Full	Full	Full	Full	n/a

5.2.7 Software Safety Design

Safety software development requires identification of hazards (i.e., abnormal conditions and events) that have the potential for defeating a safety function and implementation of design strategies to eliminate or mitigate those hazards. Software is only one component of the overall safety system. It may be embedded in instrumentation and control systems, 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 safety software, system level safety analysis is performed. The analysis should then be performed at the software component level to ensure adequate safeguards are performed to eliminate or mitigate the potential occurrence of a software defect that could cause a system failure. Methods to mitigate the consequences of safety software problems should be an integral part of the software design.[41] Specific software analysis and design methods for ensuring that safety functions are well thought out and addressed properly can 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 safety software, potential failures need to be identified, evaluated for their consequences of failure, and probability of occurrence. 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 mode and effects analysis (FMEA), fault-tree modeling, event-tree modeling, cause-consequence diagrams, hazard and operability (HAZOP) analysis, and interface analysis, and can be applied 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.[42] 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 non-safety modules, minimizing the impact of failure of one module on another.[43] 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 non-safety components. Software engineering practices that include process flow analysis, data flow analysis, path analysis, interface analysis, and interrupt analysis are techniques that are appropriate to be applied during the design phase.

When hazards related to software functions can not be eliminated, the hazard should be reduced and/or monitored. Additionally, safety 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

[41] Page 106, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part II Subpart 2.7 Quality Assurance Requirements for Computer Software for Nuclear Facility Applications,* Section 402, American Society of Mechanical Engineers, New York, New York, 2001.

[42] Page 400-412, *Safeware*, Nancy Leveson, Addison Wesley, 1995.

[43] Page 13, IEEE 7-4.3.2 *IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations Section 5.6 Independence*, Institute of Electrical and Electronic Engineers, Inc., 2003.

functions, n-version programming, and Petri nets are examples of techniques[44,45] 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 PROM checksums and watch dog timers for software processes, including operating system processes.[46] Additionally, safety 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 design work activity for Level A custom developed, configurable, and utility calculational safety software should fully meet this requirement. For this software source 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 present an unacceptable risk to failure of the safety system, fault tolerant and self-diagnostics designs should be implemented.

For custom developed, configurable, and utility calculational Level B safety software may be graded for this work activity. This grading should include fully performing the safety analysis activities for the software components. The design concepts of simplicity and isolation, as well as, fault tolerance and self-diagnostics may not apply to Level B safety software and thus, can optionally be applied.

This work activity does not apply to acquired or commercial design and analysis safety software source types.

5.2.8 Verification and Validation

Verification and validation (V&V) is the largest area within the SQA practices. Verification is performed throughout the lifecycle of the safety software. Validation activities are performed at the end of the software development or acquit ion processes to ensure the software meets the intended requirements. V&V activities should be performed by competent staff other than those whom developed the item being verified or validated.[47] V&V activities include reviews, inspections, assessments, observations, and testing. This Guide expands ASME NQA-1's

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

[45] Appendix C, *Surface Vehicle/Aerospace Recommended Practice-Software Reliability Program Implementation Guide*, Society of Automotive Engineers, SAE JA1003, January 2004.

[46] Page 13, IEEE 7-4.3.2 *IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations Section 5.5.3 Fault Detection and Self-Diagnostics*, Institute of Electrical and Electronic Engineers, Inc., 2003.

[47] Page 16, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part I Requirement 3Design Control,* Section 801.1, American Society of Mechanical Engineers, New York, New York, 2001.

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,[48] software design, code modules, test results, training materials, user documentation, and processes that guide the software development activities can 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.[49] 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 safety software for use.[50] This verification may be performed as part of the software development and implementation activity.

Assessments are important aspects of V&V. Assessments are covered in Section 5.2.4 Procurement and Supplier Management and Section 5, Assessment and Oversight.

Observations and testing can be performed during the development, factory or site acceptance, installation, and operation (aka in-use testing)[51] of the safety software. Observations and testing during development is discussed in Section 5.2.6, Software Design and Implementation. Safety 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.[52]

Acceptance testing could 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. The testing strategies, such as equivalence class testing, branch and path testing, statistical-based and boundary value testing, are appropriate for acceptance testing.

[48] Page 18, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part I Requirement 4 Procurement Document Control,* Section 300, American Society of Mechanical Engineers, New York, New York, 2001.

[49] Page 106, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part II Subpart 2.7 Quality Assurance Requirements for Computer Software for Nuclear Facility Applications,* Section 402.1, American Society of Mechanical Engineers, New York, New York, 2001.

[50] Page 106, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part II Subpart 2.7 Quality Assurance Requirements for Computer Software for Nuclear Facility Applications,* Section 402.1, American Society of Mechanical Engineers, New York, New York, 2001.

[51] Page 29 , ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part I Requirement 11 Test Control,* Section 400, American Society of Mechanical Engineers, New York, New York, 2001.

[52] Page 29, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part I Requirement 11 Test Control,* Section 200, American Society of Mechanical Engineers, New York, New York, 2001.

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.[53] If testing is not possible, monitoring using quantitative measurements should be performed.

When a new version of a software product is obtained, the site should perform predetermined and ad-hoc test cases and procedures to validate that the system meets the requirements and does not perform any unintended functions.[54] If system is operational, only positive testing may be possible.

This work activity applies to custom developed, configurable, acquired and utility calculations. 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 for safety software 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 Level A safety software, acceptance testing work activities should be planned and documented. Additionally for Level A, 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 custom developed, configurable, acquired and utility calculations can use a graded approach by applying less formality in documenting the acceptance test planning activities, test cases and procedures. 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 either Level A or Level B, when new releases of the safety software have been developed, reviews and acceptance testing of changed documents and software should be performed according to the above grading.

5.2.9 Problem Reporting and Corrective Action

Coupled with the configuration management of the safety software system, the problem reporting and corrective action process should address the appropriate requirements of the QA program corrective action system. The reporting and corrective action system will cover: (1) methods for documenting, evaluating and correcting software problems; (2) an evaluation

[53] Page 30, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part I Requirement 11 Test Control,* Section 400, American Society of Mechanical Engineers, New York, New York, 2001.

[54] Page 106, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part II Subpart 2.7 Quality Assurance Requirements for Computer Software for Nuclear Facility Applications,* Section 404, American Society of Mechanical Engineers, New York, New York, 2001.

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.[55] 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 the users should be apprised to ascertain any impacts upon safety basis decisions.

Procurement documents should identify the requirements for vendors to report problems to the supplier, any required supplier response, and the method for the purchasers to report problems to the supplier.[56]

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.[57] This work activity should be performed for Level A and B software source types: custom developed, acquired and configurable. A graded approach that reduced the formality of documenting problem reports and corrective actions taken may be applied for Level A and B utility calculation safety software.

5.2.10 Training of Personnel

Training of personnel either developing or using the safety software application is critical for minimizing the consequences of failure. Although the software testing 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. The analyst may elect to include additional defense in depth or fault tolerant technologies to reduce the operational risks. In addition, training is a proactive technique to minimize the risk of improper or invalid use of the safety software application.

Training may be necessary for both the analyst, development team and the application users. The analyst and developers may need training in fault tolerant methodologies, agent technologies, safety design methodologies, user interface design issues, testing methodologies or configuration management to ensure delivery of a robust safety software application. Meanwhile, the application user or analyst may need application specific training and safety.

[55] Page 105, , ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part II Subpart 2.7 Quality Assurance Requirements for Computer Software for Nuclear Facility Applications,* Section 204, American Society of Mechanical Engineers, New York, New York, 2001.

[56] Page 105, , ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part II Subpart 2.7 Quality Assurance Requirements for Computer Software for Nuclear Facility Applications,* Section 301, American Society of Mechanical Engineers, New York, New York, 2001.

[57] Page 229, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part IV, Subpart 4.1Guide on Quality Assurance requirements for Software,* Section 204, American Society of Mechanical Engineers, New York, New York, 2001.

Training should be commensurate with the scope, complexity, and importance of the tasks and the education, experience, and proficiency of the person. Indoctrination[58] as described by ASME NQA-1-2000 meets this work activity requirement. Moreover, personnel should 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.[59]

This work activity has no grading associated with its performance. Both Level A and Level B and custom developed, configurable, acquired, and utility calculational software source types of safety software should fully meet this requirement. This work activity does not apply to commercial design and analysis safety software.

6. ASSESSMENT AND OVERSIGHT

6.1 GENERAL

DOE assessment requirements of the QA Rule 10 CFR 830 and Order 414.1C should be applied to safety software management and control issues. DOE G 414.1-1, contains guidance on independent and management assessment.

6.2 DOE AND CONTRACTOR ASSESSMENT

DOE should assess the effectiveness of its actions in resolving issues related to safety software management and controls and evaluate the adequacy and implementation effectiveness of contractor's 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; DOE P 450.5, Line ES&H Oversight Policy*, 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, design of structures, systems and components (SSCs), and instrumentation and controls (I&C) in the DOE's defense nuclear facilities.

The organization responsible for the work will assure that the SQA implementation process meets the requirements of this guide throughout the entire software life cycle.

[58]Page 10, ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part I Requirement 2Quality Assurance Program,* Section 200, American Society of Mechanical Engineers, New York, New York, 2001.

[59]DOE-STD-1172-2003*, Safety Software Quality Assurance Functional Area Qualification Standard*, December 2003.

6.3 DOE INDEPENDENT OVERSIGHT

The DOE Office of Oversight and the OIG are responsible for conducting independent oversight of DOE actions related to Safety software issues.

The DOE/NNSA SQA responsible person at each location will verify that the SQA implementation process meets the requirements of this guide throughout the entire software life cycle.

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
CFR	Code of Federal Regulations
CIO	Chief Information Officer
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integrated
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
EH	Office of Environment, Safety and Health
EMI	Electromagnetic Interference
EPRI	Electric Power Research Institute
FMEA	Failure Modes and Effects Analysis
GAO	General Accounting Office
HAZOP	Hazards and Operability Analysis
HMI	Human-Machine Interface
HVAC	Heating, Ventilation, and Air Conditioning
I&C	Instrumentation and Controls
IAEA	International Atomic Energy Agency
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IP	Implementation Plan
ISO	International Organization for Standardization
LAN	Local Area Network
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 Plan
QARD	Quality Assurance Requirements Document
RSICC	Radiation Safety Information Computational Center

SAE	Society of Automotive Engineers
SAR	Safety Analysis Report
SC	Safety Class
SCADA	Supervisory Control and Data Acquisition
SCM	Software Configuration Management
SDD	Software Design Description
SEI	Software Engineering Institute
SG	Safety Guide
SMS	Safety Management System
SPMP	Software Project Management Plan
SQA	Software Quality Assurance
SQAIP	Software Quality Assurance Implementation Plan
SQAP	Software Quality Assurance Plan
SRS	Software Requirement Specification
SS	Safety-Significant
SSC	Structure, System, and Component
swCMM	software Capability Maturity Model
TR	Technical Report
TSR	Technical Safety Requirement
UK	United Kingdom
UL	Underwriters Laboratory
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.

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 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 non-radiological hazards, and
- any other relevant factors.

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.

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: Appendix C4.6, *Surface Vehicle/Aerospace Recommended Practice-Software Reliability Program Implementation Guide, Risk Management*, Society of Automotive Engineers, SAE JA1003, January 200410 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 Analysis & Design Software. Computer software that is not part of an SSC but is used in the safety classification, design, and analysis of nuclear facilities to ensure the proper accident analysis of nuclear facilities; the proper analysis and design of safety SSCs; and the proper identification, maintenance, and operation of safety SSCs. Source: DOE Implementation Plan for DNFSB Recommendation 2002-1 and 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 Software. Includes both safety system software and safety analysis and design software. Source: DOE Implementation Plan for DNFSB Recommendation 2002-1 and DOE O 414.1C.

Safety SSCs. The set of safety-class structures, systems, and components, and safety significant structures, systems and components for a given facility. Source: 10 CFR 830.

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. Computer software and firmware that performs a safety system function as part of a structure, system, or component (SSC) that has been functionally classified as safety class (SC) or safety significant (SS) as per 10 CFR 830.2. Safety system software includes human-machine interface software, network interface software, and programmable logic controller (PLC) programming language software. Safety system software also includes safety management databases that are not part of an SSC but whose operation or malfunction can directly affect SS and SC SSC function. Source: DOE Implementation Plan for DNFSB Recommendation 2002-1 and DOE O 414.1C.

Service. Work, such as design, construction, fabrication, decontamination, environmental remediation, waste management, laboratory sample analysis, safety software development/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, operating systems, procedures, and associated documentation and data pertaining to the operation of a computer system. Source: Institute of Electrical and Electronics Engineers (IEEE) Std. 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.

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, 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

The development and maintenance of a collection, or "toolbox," of multiple-site use, standard solution, SQA-compliant safety software, is one of the improvement actions identified by DOE for safety software. The purpose of this appendix is to outline the procedure for adding new software to the DOE SQA Central Registry that is consistent with SQA requirements of DOE Order 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 the website, 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 DOE 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 perform the major tasks. Included are the:

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 toolbox, 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] U.S. Department of Energy, *Quality Assurance*, DOE O 414.1C.

[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, (March 13, 2003).

SQA Evaluator - an independent reviewer of the computer software, who is not affiliated with the software developing organization. It is required the review organization or individuals have a thorough understanding of 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 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 toolbox 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 safety-related software application for consideration as toolbox-equivalent is a two-phase documentation effort, consisting of a 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:

(a) Widespread use of the software across the DOE Complex for safety related applications

(b) Lack of a central SQA entity to ensure proper software information, error reporting, configuration control and other SQA management

(c) 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 to the Central Registry, and is performed by the Software Evaluator. The evaluation should demonstrate satisfactory compliance with toolbox software criteria and requirements established for review of the initial set of software applications designated for the

DOE Central Registry, and performed to the same level of detail. 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 Library under the SQA Central Registry website, (*Software Information Template*, http://www.eh.doe.gov/sqa/doc_library.htm). A condensed version of the Software Information Template is shown as Attachment 1 to this appendix.

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

1. Software Project Management and Software Quality Assurance Plans
2. Software risk management documents
3. Software configuration management plan
4. Procurement and vendor 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 Manual, and Training Package/User Qualification documents

Files, reports, telephone conferences, and other documented communications can provide confirmatory indications that actions have been performed in a SQA program, 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 Toolbox. The details associated with each topical area are discussed in detail in the SQA plan and criteria document at the SQA Central Registry website.[3]

Because current and potential Central Registry software is best described under the custom category, requirements for evaluation of software should be consistent with the grading approach for custom software. Table B-2 lists SQA work activities discussed in Section 5 of this guide for custom 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 practices, 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, (November 2003).

Table B-1. SQA Topical Area and Corresponding Documentation for Demonstrating Compliance

DOE G 414.1C Software Practice	SQA Document(s)
1. Software Project Management and Quality Planning	− Software Project Management Plan (SPMP) and/or − Software Quality Assurance Plan (SQAP)
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 Vendor 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 Document (SDD); Model Description; Programmer's Reference Manual, or other related documents
7. Software Safety Design	− Software Design Document (SDD); − (Sections in other documents)
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	− User Instructions or User's Manual; − Training Package 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 of: *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 practices, 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 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 and is contained as Attachment 1 to this appendix.

While grading level B cases can be postulated, it is believed that most software applications that may be candidates for the Central Registry will best fit under grading level A.

The SQA evaluation (gap analysis) reports performed on the six designated 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 Central Registry 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 website (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. SQA Requirements by Software Grading Level and Matching DOE O 414.1C Software Practices

DOE O 414.1C Software Practice(s)*	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	Grade***	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 design	Full	Grade	6. Design
(h) Verification and validation	Full	Graded	8. Testing 10. Acceptance Test 11. Operation and Maintenance
(i) Problem reporting and corrective action	Full	Full	13. Error Impact
(j) Training of personnel	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 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	a. Determine that sufficient information is provided by the software developer to allow proper classification of the software. b. Review Developer reports, previous evaluations, and conference and journal submittals, etc. c. Interview Software Developer.
2. Evaluate Justification (Rationale) for Including Software in Central Registry	Review Software Sponsor's document: a. Widespread use of the software across DOE Complex for safety related applications? b. Lack of a central SQA entity to ensure proper software information, error reporting, configuration control and other SQA management? c. Demonstrated and quantifiable benefit for designating the software for the Central Registry?
3. Process Software Information Template	Reference Attachment 1 to this appendix, or download template from SQA website. Confirm Graded Level Determination.
4. Assess Software Project Management and Software Quality Assurance Plans	a. Review SPMP and SQAP for: • Required activities, documents, and deliverables. • 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. b. Review engineering documentation identified in the SPMP and SQAP, including: • Software risk management documents, • Software configuration management plan, • Procurement and vendor 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 Manual, and Training Package/User Qualification documents.

Step	Procedure
5. Assess Software Work Practices	Review Software Quality Assurance Documentation against detailed criteria found in the Software Evaluation Template for DOE O 414.1C Work practices: • 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 design, • Verification and validation, • Problem reporting and corrective action, and • Training of personnel.
6. Document Evaluation Using Software Evaluation Template.	Use gap analysis reports as examples.

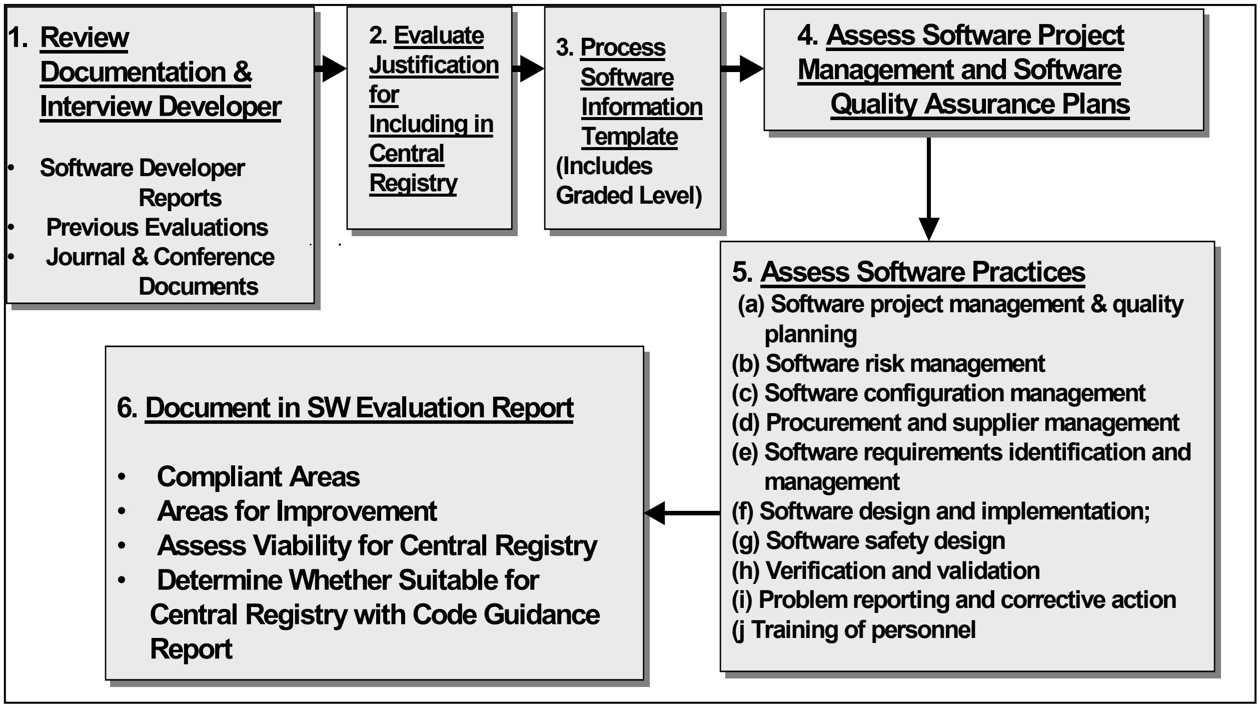
1. Review Documentation & Interview Developer
• Software Developer Reports
• Previous Evaluations
• Journal & Conference Documents
2. Evaluate Justification for Including in Central Registry
3. Process Software Information Template (Includes Graded Level)
4. Assess Software Project Management and Software Quality Assurance Plans
5. Assess Software Practices
(a) Software project management & quality planning
(b) Software risk management
(c) Software configuration management
(d) Procurement and supplier management
(e) Software requirements identification and management
(f) Software design and implementation;
(g) Software safety design
(h) Verification and validation
(i) Problem reporting and corrective action
(j Training of personnel
6. Document in SW Evaluation Report
• Compliant Areas
• Areas for Improvement
• Assess Viability for Central Registry
• Determine Whether Suitable for Central Registry with Code Guidance Report

Figure B-1. Flow Sheet for Software Evaluation

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 eleven topical areas, 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, 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 website (http://www.eh.doe.gov/sqa/doc_library.htm).

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 provide for the software in question, and a general notice will be posted on the Central Registry website. Other, 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. Lack of a central SQA entity to ensure proper software information, error reporting, configuration control and other SQA management. c. Demonstrated and quantifiable benefit for designating the software to the Central Registry.
2. SQA Technical Basis	a. The Software Quality Assurance 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 Software Quality Assurance 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 (site user or software developer) 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 eleven topical areas, 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/EH 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 website 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:

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 another software application(s) is being used for the specified safety applications.
3. The DOE/EH 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 website for a comment period of sixty 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 CONTINUED USE OF OLDER SOFTWARE VERSIONS OR RETIRED SOFTWARE

As software applications are updated to new versions to the Central Registry, many sites may still be applying older versions. Each site using an older software version should evaluate its application of the software and determine the impact of the change from the current to the newer version. The site should then document a technical basis for applying the newer version of the software, or retaining the current one, per applicable site/laboratory procedures. If the decision is made to retain the older software version, the documentation should address why the current safety software and its application are still acceptable.

There can also be situations where older software is retired and is no longer maintained in the Central Registry, but still warrants application at a specific DOE site. The same case-by-case evaluation should be performed and documented as was described above for new software versions.

Each site should periodically review changes to the Department's Central Registry, and determine if its safety applications should be updated to new software versions or software. At minimum, it is suggested that this review be performed in conjunction with the annual update cycle associated with safety basis documentation.

B.3. BIBLIOGRAPY

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. American Society of Mechanical Engineers NQA-1-2000, *Foreword to Quality Assurance Requirements for Nuclear Facility Applications* (2000).

3. ASME NQA-1a-1999, Addenda to ASME NQA-1-1997 Edition, *Quality Assurance Requirements for Nuclear Facility Applications; ASME NQA-1-2000, Quality Assurance Requirements for Nuclear Facility Applications*.

4. Code of Federal Regulations (CFR). 10 CFR 830, *Nuclear Safety Management Rule*.

5. Code of Federal Regulations (CFR). 10 CFR 830, Nuclear Safety Management Rule, Subpart A, Quality Assurance Requirements.

6. Code of Federal Regulations (CFR). 10 CFR 50, Appendix B, *Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants*.

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

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

9. International Organization for Standardization, ISO 9001-1994, *Quality systems -- Model for quality assurance in design, development, production, installation and servicing;*

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*, Letter, (March 28, 2003).

11. U.S. Department of Energy, Preparation Guide for *U.S. Department of Energy Nonreactor Nuclear Facility Safety Reports*, DOE-HDBK-3010-94, Change Notice 2 (April 2002).

SOFTWARE INFORMATION TEMPLATE

Submittal of Safety Software Applications for Central Registry Consideration

Information in the each of the following tables should be completed to the level that is meaningful – enter N/A if not applicable.

All information should be completed for new and revised software (i.e., a new version superseding previous versions). If the software is being revised, indicate this revision in Table 2 (under Version of the Code).

Table 1. Contact Information for Candidate Software Sponsor and Software Evaluator

Software Sponsor Organization: Point(s) of contact: Name: Telephone: Email: Fax:	
Software Evaluator Organization: Point(s) of contact: Name: Telephone: Email: Fax:	

Table 2. Summary Description of Subject Software

Table 2. Summary Description of Subject Software	
Type	**Specific Information**
Code Name	
Version of the Code: (Note if this is a revision to previously designated DOE Toolbox safety-related software).	
Developing Organization and Sponsor Information	
Auxiliary Codes	
Software Platform/Portability	
Coding and Computer(s)	
Technical Support Point of Contact	
Code Procurement Point of Contact	
Code Package Label/Title	

Table 2. Summary Description of Subject Software	
Type	**Specific Information**
Contributing Organization(s)	
Recommended Documentation - Supplied with Code Transmittal upon Distribution or Otherwise Available	
Input Data/Parameter Requirements	
Summary of Output	
Nature of Problem Addressed by Software	
Significant Strengths of Software	

Table 2. Summary Description of Subject Software

Type	**Specific Information**
Known Restrictions or Limitations	
Preprocessing (set-up) time for Typical Safety Analysis Calculation	
Execution Time	
Computer Hardware Requirements	
Computer Software Requirements	
Other Versions Available	

Table 3. Graded Level Determination

Basis for this graded level	

1. Software Project Management and Quality Planning

The software project management plan and software quality assurance plan for your software may be standalone documents, or embedded in other documents, related procedures, QA assessment reports, test reports, problem reports, corrective actions, supplier controls, and training packages.

1.a For this software, identify the governing Software Project Management Plan (SPMP)?
[Please submit a PDF of the SPMP]

1.b For this software, identify the governing Software Quality Assurance Plan (SQAP)?
[Please submit a PDF of the SQAP, or send hard copy of the SQAP]

1.c What software quality assurance industry standards are met by the SPMP and SQAP?

1.d What federal agency standards were used, if any, from the sponsoring organization?

1.e Has either the SPMP or SQAP been revised since the current version of the Subject Software was released? If so, what was the impact to the subject software?

1.f Are the SPMP and SQAP proceduralized in your organization? If so, please list the primary procedures that provide guidance.

Guidance for SQA Plans:

Requirement 2 – SQA Procedures/Plans (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)
ASME NQA-1 2000 Section 200
IEEE Standard 730, *IEEE Standard for Software Quality Assurance Plans*
IEEE Standard 730.1, *IEEE Guide for Software Quality Assurance Planning*

2. Software Risk Management

Software risk management provides a disciplined environment for proactive decision-making to assess continuously what can go wrong, determine what risks are important to deal with, and implement actions to deal with those risks. Because risk management is such a fundamental tool for project management, it is an integral part of software project management. 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 then 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.

How are risks managed for the subject software? How is this documented?

Guidance for Risk Management:

IEEE Standard 730, *IEEE Standard for Software Engineering: Software Life Cycle Processes, Risk Management." ISO/IEEE Standard 16085*
Software Quality Assurance Subcommittee, SQAS21.01.00-1999, *Software Risk Management: A Practical Guide*

3. Software Configuration Management

A process and related documentation for (Software Configuration Management) SCM should be defined, maintained, and controlled.

The appropriate documents, such as project procedures related to software change controls, should verify that a software configuration management process exists and is effective.

The following points should be covered in SCM document(s):

- A Software Configuration Management Plan, either in standalone form or embedded in another document,
- Configuration management data, such as software source code components, calculational spreadsheets, operational data, run-time libraries, and operating systems,
- A configuration baseline with configuration items that have been placed under configuration control,
- Procedures governing change controls,
- Software change packages and work packages to demonstrate that (1) possible impacts of software modifications are evaluated before changes are made, (2) various software system products are examined for consistency after changes are made, and (3) software is tested according to established standards after changes have been made.

3.a **Has a Software Configuration Management Plan been prepared, or are its constituent parts covered elsewhere?** [If available, submit a PDF of the Software Configuration Management Plan and related procedures].

3.b **Identify the process and procedures governing control and distribution of the subject software with users.**

3.c Do you currently interact with a software distribution organization, such as the Radiation Safety Information Computational Center (RSICC)?

3.d A Central Registry organization, under the management and coordination of the Department of Energy's Office of Environment, Safety and Health (EH), will be responsible for the long-term maintenance and control of the safety analysis toolbox codes for DOE safety analysis applications. Indicate any questions, comments, or concerns on the Central Registry's role and the maintenance of the subject software.

Guidance for Software Configuration Management Plan Documentation**:**

Requirement 12 – *Configuration Control* - SQA Procedures/Plans (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)
ASME NQA-1 2000 Section 203
IEEE Standard 828, *IEEE Standard for Software Configuration Management Plans*

4. Procurement and Supplier Management

As noted in Section 5.2.5 of DOE G 414.1C, software procurement contracts and other procurement documentation should include the technical and quality requirements for the safety software. These requirements should be verified for completeness and to assess the quality of the safety software being purchased.

There are four major approaches for this assessment: (1) performing an assessment of the supplier, (2) requiring the supplier to provide a self-declaration that the safety software meets the intended quality, (3) verifying the supplier has obtained a certification of the safety software quality from a 3rd party, and (4) accepting the safety software based upon key characteristics (e.g., large user base).

4.a How was the subject software obtained? Indicate whether it was distributed by a software center (or other third party), procured directly from the software developer, or obtained in some other way.

4.b Which of the four approaches summarized above was used to assess the technical and quality requirements for the subject software?

4.c Is this assessment documented? If so, a PDF of the documentation is requested.

Guidance for Procurement and Supplier Management:

ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications*, *Part I Requirement 4 Procurement Document Control,* Section 202, American Society of Mechanical Engineers, New York, New York, 2001
ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part I Requirement 4 Procurement Document Control,* Section 100*,* Section 802
ASME NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part II Subpart 2.7 Quality Assurance Requirements for Computer Software for Nuclear Facility Applications,* Section 301

5. Software Requirements Identification and Management

The software requirements specification (SRS) and related documentation should contain functional and performance requirements for the subject software. It may be contained in a standalone document or embedded in another document, and should address functionality, performance, design constraints, attributes and external interfaces.

5.a **For this software, was a software requirements description documented with the software sponsor?** [If available, please transmit a PDF of the Software Requirements Description, or if not available, transmit a paper copy].

5.b **If a SRS was not prepared, are there written communications that indicate agreement on requirements for the software? Please list other sources of this information if not available in one document.**

Guidance for Software Requirements Documentation:

Requirement 5 – SQA Procedures/Plans (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)
ASME NQA-1 2000 Section 401
IEEE Standard 830, *Software Requirements Specifications*

6. Software Design and Implementation

Software design documentation (SDD) and related information depict how the software is structured to satisfy the requirements from the software requirements description. It should be defined and maintained to ensure that software will serve its intended function. The SDD for the subject software may be contained in a standalone document or embedded in another document.

The SDD should provide the following:

- Description of the major components of the software design as they relate to the software requirements,

- Technical description of the software with respect to the theoretical basis, mathematical model, control flow, data flow, control logic, and data structure,
- Description of the allowable or prescribed ranges of inputs and outputs,
- Design described in a manner suitable for translating into computer coding, and
- Computer program listings (or suitable references).

6.a For the subject software, was a software design document prepared, or were its constituents parts covered elsewhere? [If available, please submit a PDF of the Software Design Document, or include hard copy with transmittal of SQAP]

6.b If the intent of the SDD information is satisfied in other documents, provide the appropriate references (document number, section, and page number).

Guidance for Software Design Documentation:

Requirement 6 – SQA Procedures/Plans (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)
ASME NQA-1 2000 Section 402
IEEE Standard 1016.1, *IEEE Guide for Software Design Descriptions* IEEE Standard 1016-1998, *IEEE Recommended Practice for Software Design Descriptions*
IEEE Standard 1012, *IEEE Standard for Software Verification and Validation;*
IEEE Standard 1012a, *IEEE Standard for Software Verification and Validation – Supplement to 1012*

7. Software Safety Design

Safety software development requires identification of hazards (i.e., abnormal conditions and events) that have the potential for defeating a safety function and implementation of design strategies to eliminate or mitigate those hazards.

7.a For this software, was an attempt made to identify potential hazards in the design phase that could defeat a safety function? What technique or approach is used? If this work practice is judged to not be applicable to the candidate software, enter "N/A ."

7.b How were these hazards mitigated or eliminated? What reports and documentation are available that describe this approach?

Guidance for Software Safety Design:

ASME NQA-1-2000, Quality Assurance Requirements for Nuclear Facility Applications, Part II Subpart 2.7 Quality Assurance Requirements for Computer Software for Nuclear Facility Applications, Section 402, American Society of Mechanical Engineers, New York, New York, 2001
IEEE 7-4.3.2 *IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations Section 5.6 Independence*, Institute of Electrical and Electronic Engineers, Inc., 2003

8. Verification &Validation

Verification and Validation (V&V) documentation should confirm that a software V&V process has been defined, that V&V has been performed, and that related documentation is maintained to ensure that:

The software adequately and correctly performs all intended functions, and the software does not perform any unintended function.

The software V&V documentation, either as a standalone document or embedded in other documents and should describe:

- The tasks and criteria for verifying the software in each development phase and validating it at completion,
- Specification of the hardware and software configurations pertaining to the software V&V,
- Traceability to both software requirements and design,
- Results of the V&V activities, including test plans, test results, and reviews (also see 5.b below),
- A summary of the status of the software's completeness,
- Assurance that changes to software are subjected to appropriate V&V,
- V&V is complete, and all unintended conditions are dispositioned before software is approved for use, and
- V&V performed by individuals or organizations that are sufficiently independent.

8.a For the subject software, identify the V&V Documentation that has been prepared. [If available, please submit a PDF of the Verification and Validation Documentation, or include a hard copy with transmittal of SQAP]

8.b If the intent of the V&V Documentation information is satisfied in one or more other documents, provide the appropriate references (document number, section, and page number). For example, a "Test Plan and Results" report, containing a plan for software testing, the test results, and associated reviews may be published separately.

Additional Testing Documentation

Test case description and outcome documentation should confirm that a software testing process has been defined, that testing has been performed, and that related documentation is maintained. Due to the overlap of testing and the verification and validation phase of work, much of the information describing testing and its outcome may be included in part or in whole with Verification and Validation documentation.

8.c If the intent of Test Case and Outcome Documentation information is satisfied in one or more other documents, provide the appropriate references (document number, section, and page number). For example, a "Verification and Validation" report, containing a plan for software testing, the test results, and associated reviews may have been published that contains sufficient information on the testing of the software.

8.d Testing of software: What has been used to test the subject software?

□ Experimental data or observations
□ Standalone calculations
□ Other validated software
□ Software is based on previously accepted solution technique

Provide any reports or written documentation substantiating the responses above.

Guidance for Software Verification & Validation Documentation:

Requirement 6 – *Design Phase* - SQA Procedures/Plans (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)
Requirement 8 – *Testing Phase* - SQA Procedures/Plans (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)
Requirement 10 – *Acceptance Test* - SQA Procedures/Plans (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)
ASME NQA-1 2000 Section 402 (Note: Some aspects of verification may be handled as part of the Design Phase)
ASME NQA-1 2000 Section 404 (Note: Aspects of validation may be handled as part of the Testing Phase)
IEEE Standard 1012, *IEEE Standard for Software Verification and Validation*
IEEE Standard 1012a, *IEEE Standard for Software Verification and Validation – Supplement to 1012*
IEEE Standard 829, *IEEE Standard for Software Test Documentation*
IEEE Standard 1008, *Software Unit Testing*

9. Problem Reporting and Corrective Action

Software problem reporting and corrective action documentation help ensure that a formal procedure for problem reporting and corrective action development for software errors and failures is established, maintained, and controlled.

A Software Error Notification and Corrective Action report, procedure, or similar documentation, should be implemented to report, track, and resolve problems or issues identified in both software items, and in software development and maintenance processes. Documentation should note specific organizational responsibilities for implementation. Software problems should be promptly reported to affected organizations, along with corrective actions. Corrective actions taken ensure that:

- Problems are identified, evaluated, documented, and, if required, corrected,
- Problems are assessed for impact on past and present applications of the software by the responsible organization,
- Corrections and changes are executed according to established change control procedures, and
- Preventive actions and corrective actions results are provided to affected organizations.

9.a **Identify documentation specific to the subject software that controls the error notification and corrective actions.** [If available, please submit a PDF of the Error Notification and Corrective Action Report documentation for the subject software (or related procedures). If this is not available, include hard copies with transmittal of SQAP].

9.b **Provide examples of problem/error notification to users and the process followed to address the deficiency. Attach files as necessary.**

9.c **Provide an assessment of known errors or defects in the subject software and the planned action and time frame for correction.**

Category of Error or Defect	Corrective Action	Planned schedule for correction
Major		

Category of Error or Defect	Corrective Action	Planned schedule for correction
Minor		

9.d Identify the process and procedures governing communication of errors/defects related to the subject software with users.

Guidance for Error Notification and Corrective Action Documentation**:**

Requirement 13 – *Error Impact* - SQA Procedures/Plans (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)
ASME NQA-1 2000 Section 204
IEEE Standard 1063, *IEEE Standard for Software User Documentation*

10. <u>Training of Personnel</u>

Training of personnel and user instruction documentation are necessary to assist the user in correctly installing, operating, managing, and maintaining the software, and to ensure that the software satisfies user requirements. At minimum, the documentation should describe:

- The user's interaction with the software
- Any required training prerequisites
- Input and output specifications and formats, options
- Software limitations
- Error message identification and description, including suggested corrective actions to be taken to correct those errors, and
- Other essential information for using the software.

10.a For the subject software, has Software User Documentation been prepared, or are its constituents parts covered elsewhere? [If available, submit a PDF of the Software User Documentation, or include a hard copy with transmittal of SQAP]

10.b If the intent of the Software User Documentation information is satisfied in other documents, provide the appropriate references (document number, section, and page number).

10.c Is other related documentation available? This may include, but is not limited to, model description, user guide, and input/output of example cases.

10.d Are Training Package and User Qualification statements available?

10.e What training is offered to guide the user in correctly executing the subject software? Complete the appropriate section from the following:

Type	**Description**	**Frequency of training**
Training Offered to User Groups as Needed		
Training Sessions Offered at Technical Meetings or Workshops		
Training Offered on Web or Through Video Conferencing		
Other Training Modes		
Training Not Provided		

Guidance for Training of Personnel and User Documentation**:**

Requirement 9 – *User Instructions* - SQA Procedures/Plans (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)
ASME NQA-1 2000 Part 1 200 – Indoctrination and Training
Requirement 9 – SQA Procedures/Plans (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)
ASME NQA-1 2000 Section 203
IEEE Standard 1063, *IEEE Standard for Software User Documentation*

Other Supplementary Information and Planned Upgrades

This section is optional but allows the software sponsor to add any remaining information that can help inform DOE Office of Quality Assurance Programs on the value of placing the candidate software in the Central Registry. Included are miscellaneous sources of information and planned upgrades.

a. Provide any other information that may supplement information previously described that supports adding/revising/removing the intended software application relative to the Central Registry.

Supplemental/Miscellaneous Information

Plan/Document/Procedure Type	Document
1. Weekly or Monthly Reports	
2. Meeting Minutes or Internal Reports	
3. Software Distribution Center Notices	
4. Other Documentation	

Software Upgrades

b. Describe modifications planned for the subject software**.**

Technical Modifications

Priority	**Description of Change**
1.	
2.	
3.	
4.	
5.	

User Interface Modifications

Priority	**Description of Change**
1.	
2.	
3.	
4.	
5.	

Software Engineering Improvements

Priority	**Description of Change**
1.	
2.	
3.	
4.	
5.	

Other Planned Modifications

Priority	**Description of Change**
1.	
2.	
3.	
4.	
5.	

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

This appendix provides guidance on the use of ASME NQA-1-2000 and supporting standards for compliance with Department of Energy's quality assurance requirements: 10 CFR 830 Subpart A and DOE O 414.1C and their application to safety software.

C.1. PURPOSE

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

C.2. INTRODUCTION

The Department of Energy (DOE) QA requirements for activities that affect, or may affect, quality, nuclear safety or other site specified criteria are established by Rule, 10 CFR Part 830 Subpart A, dated January 10, 2001. DOE also has equivalent requirements for all other federal and contractor activities in QA Order, 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 quality assurance 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 standard is a national consensus standard and should be considered for providing the essential implementing methods for a DOE 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. 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, Part I requirements. Differences in the documents and topics that should be addressed independently of the ASME NQA-1 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 is 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 414.1A dated September 29, 2001

DOE General Requirements (Summarized)	ASME NQA-1 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

TABLE C-1

10 CFR 830 Subpart A, dated January 10, 2001

§830.121 Quality Assurance Program
DOE O 414.1A dated September 29, 2001

DOE General Requirements (Summarized)	ASME NQA-1 Requirements
	acceptance criteria for determining that prescribed results have been satisfactorily attained." A DOE QAP will need to describe how the DOE criteria are satisfied.
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 and 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 Requirements	Comments, Software Requirements & Other Standards
<u>Criterion 1 - Management/Program</u>	**<u>NQA Requirements 1 and 2</u>** 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 Requirements**	**Comments, Software Requirements & Other Standards**
<u>Criterion 2 - Management/Personnel Training and Qualification</u>	**<u>NQA Requirement 2</u>** 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 Draft Computer Software Functional Area Qualification Standard, TRNG 0040 **IAEA TR 397, 2.4** **IAEA NS-G-1.1, 4.9&10**
<u>Criterion 3 - Management/Quality Improvement</u>	**<u>NQA Requirements 2, 15, and 16</u>** 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.
(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 test. Requirement 16 includes criteria to prevent recurrence of identified problems.	

TABLE C-2

10 CFR 830 Subpart A, dated January 10, 2001

§830.122 Quality Assurance Criteria

DOE Quality Assurance Criteria	ASME NQA-1 Requirements	Comments, Software Requirements & Other Standards
(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.	
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**

TABLE C-2

10 CFR 830 Subpart A, dated January 10, 2001

§830.122 Quality Assurance Criteria

DOE Quality Assurance Criteria	**ASME NQA-1 Requirements**	**Comments, Software Requirements & Other Standards**
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 Part IV, 4.1, 203 & 405 IAEA TR 397, 3.1 & 3.2 IEEE 828-1998 & 1219-1998
(2) Identify and control items to ensure their proper use.	The ASME NQA-1 requirements satisfy this element of the DOE Criterion.	
(3) Maintain items to prevent their damage, loss, or deterioration.	The ASME NQA-1 requirements satisfy this element of the DOE Criterion.	
(4) Calibrate and maintain equipment used for process monitoring or data collection.	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 Requirements	Comments, Software Requirements & Other Standards
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
Criterion 7 - Performance/Procurement	**NQA Requirements 4 and 7** The ASME NQA-1 requirements 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 Requirements**	**Comments, Software Requirements & Other Standards**
(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
Criterion 9 - Assessment/Management Assessment	**NQA Requirement 2 and 18** The ASME NQA-1 requirements partially meet the DOE Criterion, as noted	

TABLE C-2

10 CFR 830 Subpart A, dated January 10, 2001

§830.122 Quality Assurance Criteria

DOE Quality Assurance Criteria	**ASME NQA-1 Requirements**	**Comments, Software Requirements & Other Standards**
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 the 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 Guide, G 414.1-1A, *Management Assessment and Independent Assessment Requirements of 10 CFR 830.120 and DOE- O-414.1 Quality Assurance*, in order 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 Requirements	Comments, Software Requirements & Other Standards
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 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 regulatory requirements. This report 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 quality assurance, 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 lifecycle, 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 Requirements for Nuclear Facility Applications. 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, IEEE standards (e.g., configuration management),
- Provides independence for developing and verification,
- Supports graded implementation,
- Widely used among DOE contractor QA programs, and
- 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.

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 programme;
- Appendix III: Considerations before acquisition of computerized tools;
- Appendix IV: Functions of computer program understanding and reverse engineering tools;
- 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, and IEC 1226 Nuclear Power Plants—Instrumentation and Control Systems Important for Safety—classification.

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.8. EXAMPLE APPLICATION GUIDES, FEDERAL AGENCY REQUIREMENTS & 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 was:

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

D.8.2 NATIONAL AERONAUTICS AND SPACE ADMINISTRATION (NASA)

The NASA software document is the Software Assurance Standard, NASA-STD-2201-93. 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 4- CFR 104 Rule and the WIPP QA program requirements influence many other waste generation sites across the DOE complex. The regulation adopts ASME NQA-1, 1997 and Subpart 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.

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 Attachment 1 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 assure quality.

D.10. REFERENCES

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

2. Code of Federal Regulations (CFR). 10 CFR 830, *Nuclear Safety Management Rule*.

3. CFR 10 CFR 63, *Disposal of High-Level Radioactive Wastes in A Geologic Repository at Yucca Mountain*, Nevada U.S. Nuclear Regulatory Commission.

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

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

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

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

8. DOE, U.S. Department of Energy (2003). *Implementation Plan for Defense Nuclear Facilities Safety Board Recommendation 2002-1: Quality Assurance for Safety Software at Department of Energy Nuclear Facilities,* Report, (February 28, 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.

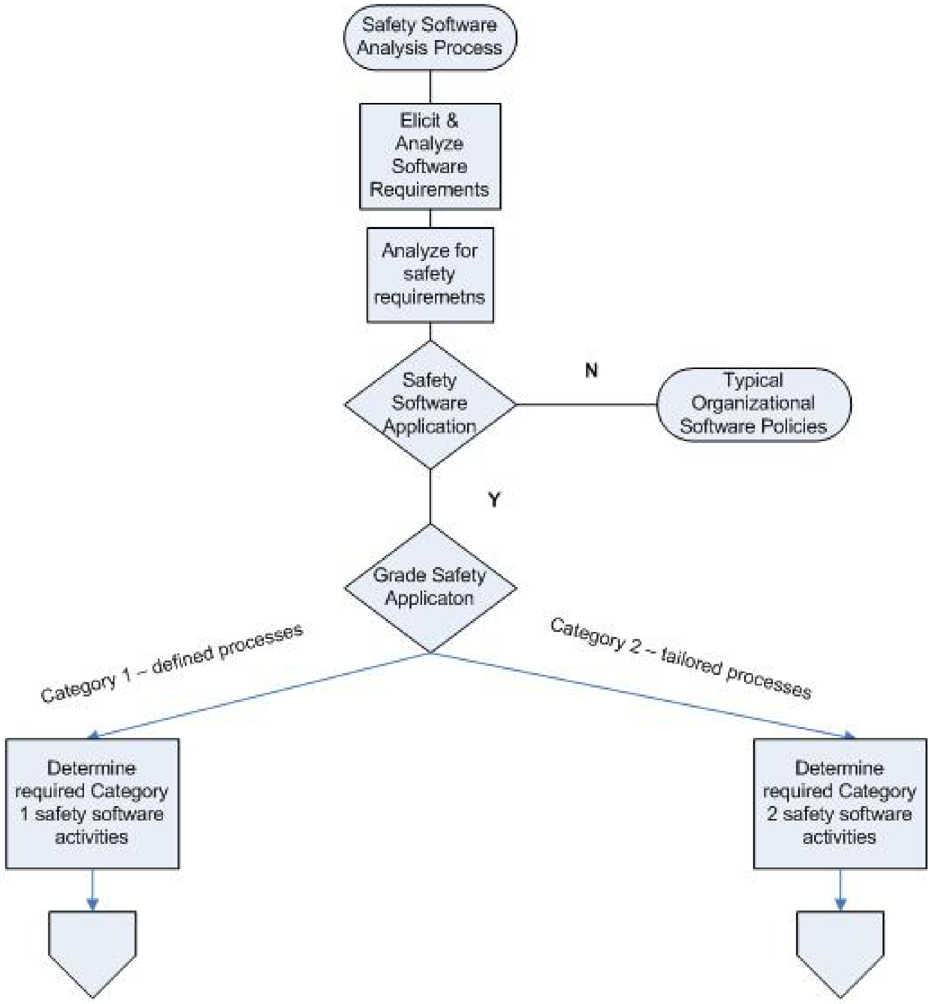
Safety Software Analysis Process
Elicit & Analyze Software Requirements
Analyze for safety requiremetns
Safety Software Application
N
Typical Organizational Software Policies
Y
Grade Safety Applicaton
Category 1 – defined processes
Category 2 – tailored processes
Determine required Category 1 safety software activities
Determine required Category 2 safety software activities

FIGURE 1: Top Level Safety Software Category Analysis Process

PLEASE NOTE: the logic diagrams provided on these two pages provide guidance regarding the analysis and tasks for the following categories of DOE safety software applications: (1) custom; (2) configurable; (3) acquired; (4) utility calculations; and (5) design and analysis.

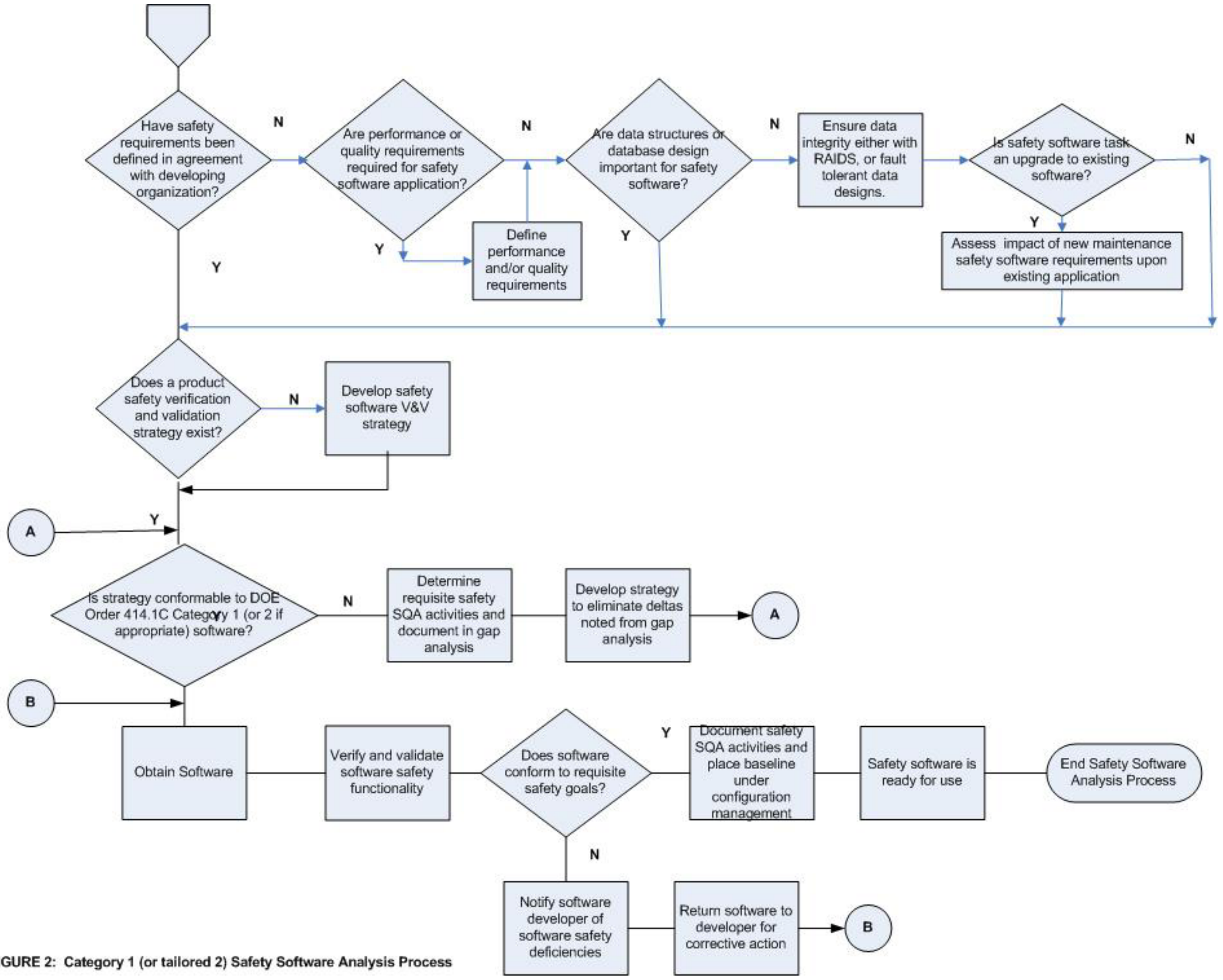
Have safety requirements been defined in agreement with developing organization?
N
Y
Are performance or quality requirements required for safety software application?
N
Y
Define performance and/or quality requirements
Are data structures or database design important for safety software?
N
Y
Ensure data integrity either with RAIDS, or fault tolerant data designs.
Is safety software task an upgrade to existing software?
N
Y
Assess impact of new maintenance safety software requirements upon existing application
Does a product safety verification and validation strategy exist?
N
Develop safety software V&V strategy
A
Y
Is strategy conformable to DOE Order 414.1C Category 1 (or 2 if appropriate) software?
Y
N
Determine requisite safety SQA activities and document in gap analysis
Develop strategy to eliminate deltas noted from gap analysis
A
B
Obtain Software
Verify and validate software safety functionality
Does software conform to requisite safety goals?
Y
Document safety SQA activities and place baseline under configuration management
Safety software is ready for use
End Safety Software Analysis Process
N
Notify software developer of software safety deficiencies
Return software to developer for corrective action
B

FIGURE 2: Category 1 (or tailored 2) Safety Software Analysis Process

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 for safety analysis, design of structures, systems and components (SSCs), and instrumentation and controls (I&C) in the 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, (2) Software Risk Management, (3) Software Requirement Description, (4) Software Design Description, (5) Software Verification and Validation, (6) Software User Documentation, (7) Software Configuration Management, (8) Software Quality Assurance, (9) Software Procurement, (10) Software Problem Reporting and Corrective Action, (11) Software Continuous Improvement, and (12) Training.
- 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 currently in use in the safety analysis, design of structures, systems and components (SSCs), and I&C in the DOE defense nuclear facilities. The scope of the assessment, henceforth, is called "Safety Software." The assessment criteria and guidelines provide a consistent framework for assessing the processes that are currently in place to ensure that the safety software in defense nuclear facilities is adequate.

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

- DOE O 414.1-C, Quality Assurance,
- American Society of Mechanical Engineers (ASME) NQA-1-2000, *Quality Assurance Requirements for Nuclear Facility Applications, Part II, Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications,*

- 10 CFR 830 Nuclear Safety Management,
- Applicable Institute of Electrical and Electronics Engineers (IEEE) standards,
- Codes of Federal Regulations (CFRs), and
- DOE directives and Guidance.

The CRADs are not prescriptive enough to evaluate the SQA programs associated with the safety software. The CRADs could be used for the assessment of following types of software:

- Custom software developed by DOE, its contractors, or subcontractors for use with safety systems or safety class structures, systems and components,
- Configurable,
- 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.

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,
- I&C software, including embedded microprocessors, distributed control systems, supervisory control and data acquisition systems (SCADAs), programmable logic controllers (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 will 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 problem. Individual sites should tailor the scope of this assessment to suit the specific usage software in their safety systems.

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 previous assessments and reviews to gather data as appropriate. 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.
- 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 team 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, V&V, testing, and maintenance should be evaluated for adequacy, and to determine whether or not 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.
- These assessment criteria and guidelines were not developed for a specific safety software application. Therefore, 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 agreed upon 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. 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.

- 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. 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 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 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 if 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 pre-assessment 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 requirement (TSR). 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 the DOE O 414.1C Quality Assurance, this Guide, and 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

[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,* 5-31-01.

2. Software Risk Management
3. Software Configuration Management
4. Software Procurement and Supplier Management
5. Software Requirements Identification and Management
6. Software Design and Implementation
7. Software Safety Design
8. Software Verification and Validation
9. Software Problem Reporting and Corrective Action
10. Training of Personnel

Each of these work activities includes:

- ***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 lifecycle 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. Software quality assurance activities, software practices, and documentation are periodically assessed.
3. Software quality activities have been effectively implemented.

Approach:

Confirm the existence of project management and quality assurance planning work activity. This may be present in software project management and/or software quality assurance plans that exist either as a standalone 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,
- Software configuration management activities,
- Software risk management approach,
- 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 verification and validation 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 verification and validation address the quality of the software verification and validation 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 software configuration management 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 verification and validation 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, and ensure that the risk management planning specifies, as applicable, 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, likelihood, and potential safety consequences 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 lifecycle, 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 establishes and implemented.
3. In addition, for Level A or Level B custom 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 a 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 after changes are made, (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 are 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 quality assurance based on risk, safety, facility lifecycle, 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’ quality assurance programs meet or exceed the quality assurance 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 quality assurance requirements. These requirements may be embedded in the DOE contractors’ or subcontractors’ procurement document, software or system design description, or a software quality assurance plan. If not documented in the procurement contract, ensure that the supplier has received such technical and quality assurance requirements. This verification may overlap with the Software Requirements Management work activity.
- Verify that the suppliers’ quality assurance program has been reviewed and meets or exceeds the procurement specification requirements. The supplier may review the supplier’s quality assurance 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 Verification and Validation 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, 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, SARs, 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 if 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.
- Verify that the software requirements are correct, unambiguous, complete, consistent, verifiable, modifiable and traceable as appropriate.

- Verify that acceptance criteria is 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 software configuration management activity.
- Verify that software requirement documents are reviewed and updated as necessary. This may overlap with the software verification and validation 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 document 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 non-safety 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 the 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 verification and validation work activity.
- Evidence of developer testing for the appropriate grading exists.

F.5.7 SOFTWARE SAFETY DESIGN

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 tolerant 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 FMEA, 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 software, and optionally for Level B safety software, sample safety software modules for proof of design complexity evaluation and isolation of safety functions from non-safety 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 should be performed by competent staff other than those whom developed the item being verified or validated. This may overlap with the training work activity.

Approach:

Review appropriate documents, such as software quality assurance 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 vendor qualification reports to determine if:

- 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 whom developed the item.
- Software design was performed prior to the safety software being used in operations.
- 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.

  —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.

- Test V&V:

  —Document for development, factory or acceptance testing, installation, and operations testing exists.

  —Test phase documentation includes test guides and the results of the execution of test cases.

—Test 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 is appropriate, to user documentation.

—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 if:

- 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.
- The organizations responsible for problem reporting and resolution are clearly defined.

F.5.10 TRAINING

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

   —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. I**ntroduction.** 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 QA Rule, may be obtained through the online DOE Directives, Regulations, and Standards web site: http://www.directives.doe.gov. Other documents, such as the ASME, ASQ, and IEC standards and guidance documents may be purchased or obtained from the sponsoring organizations.

1. 10 CFR 63 Disposal of High-Level Radioactive Wastes in a Geologic Repository at Yucca Mountain, Nevada.

2. 10 CFR 63, Disposal of High-Level Radioactive Wastes In A Geologic Repository at Yucca Mountain, Nevada U.S. Nuclear Regulatory Commission.

3. 10 CFR 830, Nuclear Safety Management.

4. 10 CFR Part 50, Appendix B, Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants.

5. 10 CFR Part 70, Domestic Licensing of Special Nuclear Material.

6. ANSI/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.

7. ANSI/ASQC E4-1994, Quality Systems for Environmental Data and Technology Programs - Requirements with Guidance for Use and latest draft revision.

8. ASME American Society of Mechanical Engineers NQA-1-2000, Foreword to Quality Assurance Requirements for Nuclear Facility Applications (2000).

9. ASME NQA-1-1997, Quality Assurance Requirements for Nuclear Facility Applications.

10. ASME NQA-1-2000, Quality Assurance Requirements for Nuclear Facility Applications.

11. ASME NQA-1a-1999, Addenda to ASME NQA-1-1997 Edition, Quality Assurance Requirements for Nuclear Facility Applications.

12. ASME NQA-3-1989, Quality Assurance Program Requirements for the Collection of Scientific and Technical Information for Site Characterization of High-Level Nuclear Waste Repositories.

13. Atomic Energy Control Board, Canada; Direction De La Sûreté Des Installations Nucléaires, Institut De Protection Et De Sûreté Nucléaire, France; Nuclear Installations Inspectorate, United Kingdom; Nuclear Regulatory Commission, United States Of America, Four Party Regulatory Consensus Report on the Safety Case for Computer-Based Systems in Nuclear Power Plants, HMSO, Norwich (1997).

14. Bishop, P. G., *Dependability of Critical Computer Systems, Techniques Directory*, Publishers Elsevier Applied Science, 1990.

15. CCPS, Guidelines for Use of Vapor Cloud Dispersion Models, Second Edition, Center for Chemical Process Safety, American Institute of Chemical Engineers, New York, NY, (1996).

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

17. DeMarco, Tom, *Controlling Software Projects*, Yourdon Press, 1982.

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

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

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

21. DNFSB Recommendation 2002-1, *Quality Assurance for Safety-Related Software.*

22. DNFSB/TECH-25, *Quality Assurance for Safety-Related Software at Department of Energy Defense Nuclear Facilities.*

23. DoD 5000.59, Modeling and Simulation (M&S) Management.

24. DoD 5000.61, Modeling and Simulation (M&S) Verification, Validation, and Accreditation (VV&A).

25. DoD-STD-882D, *Standard Practice for System Safety,* Department of Defense, 10 February, 2000.

26. DOE G 200.1-1, *DOE Guidelines for Software Engineering Methodology.*

27. 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; DOE P 450.5, Line ES&H Oversight Policy*, May 31, 2001.

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

29. DOE G 420.1-1, *Facility Safety.*

30. DOE O 200.1 Information Management.

31. DOE O 414.1A, Quality Assurance.

32. DOE O 414.1C, *Quality Assurance*.

33. DOE, *Framework for Grading Safety Software for DOE Directive Work Paper*, April 22, 2004.

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

35. DOE, U.S. Department of Energy (2003). Implementation Plan for Defense Nuclear Facilities Safety Board Recommendation 2002-1: Quality Assurance for Safety Software at Department of Energy Nuclear Facilities, Report, (February 28, 2003).

36. DOE-RW-0333P, Quality Assurance Requirements and Description for the Civilian Radioactive Waste Management Program.

37. DOE-Std-1172-2003, *Safety Software Quality Assurance Functional Area Qualification Standard.*

38. DOE-STD-3009-94, Change Notice 2, April 2002, *Preparation Guide for U.S. Department of Energy Nonreactor Nuclear Facility Documented Safety Analyses.*

39. European Nuclear Regulators' Current Requirements and Practices for the Licensing of Safety Critical Software for Nuclear Regulators, Rep. EUR 18158 EN, Office for Official Publications of the European Communities, Luxembourg (1998).

40. Herrmann, Debra, *Software Safety and Reliability*, Publishers IEEE Computer Society, 1999.

41. Huffman, Phillip, Nuclear Weapons Complex Software Quality Assurance Subcommittee, *Definitions and Analytical Techniques Supporting a Life Cycle Approach to Software Safety*, March 26, 2004, Rev 0.4.

42. IAEA Safety Guide (SG) Series No. NS-G-1.1, Software for Computer Based Systems Important to Safety in Nuclear Power Plants, Vienna (2000).

43. IAEA Safety Series No. 50-SG-D3, Protection Systems and Related and Related Systems. (1984).

44. 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, Vienna (1996).

45. IAEA Safety Series No. 50-SG-D8, Safety-Related Instrumentation and Control Systems. (1984).

46. IAEA TR Series No. 282, Manual on Quality Assurance for Computer Software Related to the Safety of Nuclear Power Plants, IAEA, Vienna (1988).

47. IAEA TR Series No. 384, Verification and Validation of Software Related to Nuclear Power Plant Instrumentation and Control, IAEA, Vienna (1999).

48. IAEA TR Series No. 397, Quality Assurance for Software Important to Safety Vienna (2000).

49. IAEA, TECDOC Series No. 1066, Specification of Requirements for Upgrades Using Digital Instrument and Control Systems, Vienna (1999).

50. IEC 12207, Information Technology – Software Life-Cycle Processes, Geneva (1995).

51. IEC 60880, Software for Computers in Safety Systems of Nuclear Power Plants, Geneva (1986).

52. IEC 60987, Programmed Digital Computers Important to Safety for Nuclear Power Stations, Geneva (1989).

53. IEC 61226, Nuclear Power Plants – Instrumentation and Control Systems Important for Safety – Classification, Geneva (1993).

54. IEC 880, *Software for computers in the safety systems of nuclear power stations,* International Electrotechnical Commission, Geneva, Switzerland, 1986.

55. IEC 9126, Information Technology - Software Product Evaluation – Quality Characteristics and Guidelines for Their Use, Geneva (1991).

56. IEEE Standard 1008-1987(R1993), Software Unit Testing.

57. IEEE Standard 1012-1998, IEEE Standard for Software Verification and Validation

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

59. IEEE Standard 1028-1997, IEEE Standard for Software Reviews.

60. IEEE Standard 1042-1987, *IEEE Guide to Software Configuration Management, Section 3.3.4 Audits and Reviews*, Institute of Electrical and Electronic Engineers, Inc., 1987.

61. IEEE Standard 1063-1987(R1993), IEEE Standard for Software User Documentation.

62. IEEE Standard 1074-1991, IEEE Standard for Developing Software Life Cycle Processes.

63. IEEE Standard 1219, *Standard for Software Maintenance.*

64. IEEE Standard 1228-1994, IEEE Standard for Software Safety Plans.

65. IEEE Standard 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology.

66. IEEE Standard 730-2002, IEEE Standard for Software Quality Assurance Plans.

67. IEEE Standard 828-1998, IEEE Standard for Software Configuration Management Plans.

68. IEEE Standard 829-1998, IEEE Standard for Software Test Documentation.

69. IEEE Standard 830-1998, Software Requirements Specifications.

70. 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, Inc., 2003.

71. IEEE/EIA Standard 12207.01996, Industry Implementation of International Standard ISO/IEC 12207 Standard for Information Technology – Software Life Cycle Processes.

72. IEEE/EIA Standard 12207.1-1997, Industry Implementation of International Standard ISO.IEC 12207 Standard for Information Technology – Software Life Cycle Processes – Life Cycle Data.

73. IEEE/EIA Standard 12207.2-1997, Industry Implementation of International Standard ISO.IEC 12207 Standard for Information Technology – Software Life Cycle Processes – Implementation Considerations.

74. IP 2000-2, *Implementation Plan for Defense Nuclear Facilities Safety Board Recommendation 2000-2, Configuration Management, Vital Safety Systems,* October 31, 2000.

75. 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*,* March 13, 2003.

76. 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.

77. ISO 9001-1994, *Quality Systems—Model for quality assurance in design, development, production, installation and servicing*, International Organization for Standardizations, 1994.

78. 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,* International Organization for Standardizations, 2000.

79. Leveson, Nancy, *Safeware*, Publishers Addison Wesley, 1995.

80. NASA-STD-2201-93*, Software Assurance*, National Aeronautics and Space Administration.

81. NASA-STD-8719.13A, *Software Safety,* National Aeronautics and Space Administration, September 15, 1997.

82. Pressman, Roger, *Software Engineering A Practitioner's Approach*, McGraw Hill, 1992

83. SAE JA1003, *Surface Vehicle/Aerospace Recommended Practice-Software Reliability Program Implementation Guide, Risk Management,* Society of Automotive Engineers, January 2004.

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

85. SQAS21.01.00, *Software Risk Management: A Practical Guide,* Department of Energy Quality Managers Software Quality Assurance Subcommittee, February 2000.