DOE-EH-4.2.1.2-Criteria

Defense Nuclear Facilities Safety Board Recommendation 2002-1
Software Quality Assurance Improvement Plan
Commitment 4.2.1.2:

Software Quality Assurance Plan and Criteria for the Safety Analysis Toolbox Codes

U.S. Department of Energy
Office of Environment, Safety and Health
1000 Independence Ave., S.W.
Washington, DC 20585-2040

November 2003

INTENTIONALLY BLANK

FOREWORD

This document discusses the Software Quality Assurance plan, and criteria and implementation procedures to be used to evaluate designated, safety-related computer software for the Department of Energy Computer Software Toolbox. The plan and criteria are intended for application to safety analysis software. The initial application will be on six computer codes used primarily for accident analysis, including ALOHA, CFAST, EPIcode, GENII, MACCS2, and MELCOR.

Suggestions for corrections or improvements to this document should be addressed to :

Chip Lagdon
EH-31/GTN
Office of Quality Assurance Programs
U.S. Department of Energy
Washington, D.C. 20585-2040
Phone (301) 903-4218
Email: Chip.Lagdon@eh.doe.gov

INTENTIONALLY BLANK

REVISION STATUS

Page/Section	Revision	Change
1. Entire Document	Interim Report	1. Original Issue
2. Entire Document	Final	2. Resolved Comments

INTENTIONALLY BLANK

Table of Contents

Section		Page
FOREWORD		iii
REVISION STATUS		v
EXECUTIVE SUMMARY		ix
1.0	INTRODUCTION	1
1.1	Designated Safety Analysis Toolbox Codes	2
1.2	Evaluation of Toolbox Codes	3
1.3	Uses of the Gap Analysis	3
2.0	SQA PLAN FOR SAFETY ANALYSIS TOOLBOX CODES	1
2.1	Software Categorization and Classification	2-3
2.2	Application of Implementation Procedures for Existing Software	2-5
3.0	SOFTWARE QUALITY ASSURANCE CRITERIA FOR SAFETY ANALYSIS TOOLBOX CODES	3-1
3.1	10 CFR 830 Subpart A	3-3
3.2	NQA-1-2000, Part II, Section 2.7 (Quality Assurance Requirements for Computer Software for Nuclear Facility Applications)	3-3
3.3	Department of Energy Order 414.1A	3-5
3.4	Implementation of Primary SQA Criteria	3-5
4.0	OVERALL PROCESS FOR SQA EVALUATION OF SAFETY ANALYSIS TOOLBOX CODES	4-1
5.0	ACRONYMS	5-1
6.0	DEFINITIONS	6-1
7.0	REFERENCES	7-1
8.0	BIBLIOGRAPHY OF STANDARDS REFERENCED IN NUCLEAR QUALITY ASSURANCE AND SOFTWARE QUALITY ASSURANCE	8-1

Appendices

Appendix A. Software Classification Levels and Graded Application A-1

Appendix B. Criteria for Software Quality Assurance for Safety-Related Software Applied to DOE Nuclear Facilities B-1

Appendix C. C-1

Appendix D. Applicable Consensus Standards and Guides D-1

Appendix E. Application to the MACCS2 Computer Code E-1

Software Quality Assurance Plan and Criteria for the Safety Analysis Toolbox Codes

Executive Summary

A plan for conducting review of the Software Quality Assurance programs, practices, and procedures for safety analysis software designated as Department of Energy toolbox codes is documented herein. The plan applies criteria based on compliance with ASME NQA-1 requirements. The present report is a deliverable under Department of Energy (DOE) Software Quality Improvement Implementation Plan, Commitment 4.2.1.2.

The primary set of baseline Software Quality Assurance criteria for evaluating designated safety analysis software for the DOE computer code toolbox are based on the following:

- American Society of Mechanical Engineers (ASME) NQA-1-2000, Subpart 2.7, and other applicable parts of the NQA-1 Standard, primarily Part I, Requirements 3 and 11
- 10 CFR 830 Nuclear Safety Management
- A consistent set of software classification criteria (Appendix A) as defined in this document.

The selection is based on:

- Nuclear industry precedent with ASME Nuclear Quality Assurance standards
- Federal and commercial sectors continued involvement with and maintenance of the ASME NQA standards
- Quality assurance perspective through connection with 10 CFR 50 Appendix B and 10 CFR 70
- Independence of roles in developing and maintaining software, among management, work performers, and work reviewers
- Graded application based on safety, risk, and hazard analysis of the function of the software
- Focus on protection of the public and workers
- Long-standing presence and incorporation with many DOE contractors' quality assurance programs, with focus on nuclear safety, and
- Completeness and relevance to scientific, applied research, design, analysis and nuclear engineering software.

Applicable Institute of Electronics and Electrical Engineers (IEEE) standards, Codes of Federal Regulations (CFRs), DOE directives and guidance, and other sections of the ASME NQA standard are used to augment the NQA-1-2000 Subpart 2.7 requirements, and will be used as supplemental criteria.

The primary SQA criteria are not prescriptive enough to evaluate the SQA programs associated with the designated safety analysis toolbox codes. Thus, a plan is provided to guide the evaluation process. The evaluation plan (Table 2-2 of this report) covers the major requirements of NQA-1-2000 with a procedural basis for evaluating software that was developed mostly outside of NQA-1 requirements, and used in accident analysis applications. The procedural basis provides instructions for evaluation of existing accident analysis software, referencing detailed criteria based on NQA-1-2000 compliant requirements (listed in Table 3-3 of this document).

The overall SQA plan for the designated toolbox codes requires the code developer organization to provide SQA documentation and other information to an independent SQA evaluator. The SQA evaluator then assesses the program, procedures, and processes associated with the software, based on review of this information, the evaluation procedure, and knowledge of the subject software.

While the plan and criteria covered herein are to be applied initially to a set of safety analysis computer software, there is sufficient flexibility to extend the proposed process to other categories of software. Included are process control and design software.

Software Quality Assurance Plan and Criteria for the Safety Analysis Toolbox Codes

1.0 Introduction

In January 2000, the Defense Nuclear Facilities Safety Board (DNFSB) issued Technical Report 25, (TECH-25), *Quality Assurance for Safety-Related Software at Department of Energy Defense Nuclear Facilities* (DNFSB, 2000). TECH-25 identified issues regarding computer software quality assurance (SQA) in the Department of Energy (DOE) Complex for software used to make safety-related decisions, or software that controls safety-related systems. Instances were noted of computer codes that were either inappropriately applied, or were executed with incorrect input data. Of particular concern were inconsistencies in the exercise of SQA from site to site, and from facility to facility, and the variability in guidance and training in the appropriate use of accident analysis software.

During the 2000 to 2002 period, survey information on SQA programs, processes, and procedures was collected by DOE from site and laboratory contractors. Initial elements for a response plan were also developed. However, to expedite implementation of corrective actions in this area, the DNFSB issued Recommendation 2002-1, *Quality Assurance for Safety-Related Software at Department of Energy Defense Nuclear Facilities*, on September 23, 2002 (DNFSB, 2002). As part of its Recommendation to DOE, the DNFSB enumerated many of the points noted earlier in TECH-25, but noted specific concerns regarding the quality of the software used to analyze and guide safety-related decisions, the quality of the software used to design or develop safety-related controls, and the proficiency of personnel using the software.

A series of actions that address the DNFSB's concerns are documented in the DOE Implementation Plan for DNFSB Recommendation 2002-1, *Implementation Plan for Defense Nuclear Facilities Safety Board Recommendation 2002-1*, (DOE, 2003). The Assistant Secretary for Environment, Safety and Health (DOE/EH) is the DOE executive responsible for the Quality Assurance Program, and DOE/EH is responsible for ensuring the successful completion of the implementation plan. The Implementation Plan (IP) was accepted by the DNFSB in April 2003 as adequately addressing the concerns raised by Recommendation 2002-1.

Section 4.2 of the IP on Computer Codes recognizes that some computer codes are widely used in the DOE Complex for safety analysis applications, or could have significant consequences in the event of failure. Responsive to the Recommendation 2002-1, the IP designates computer software of this type as toolbox codes, i.e., standard computer software meeting minimum SQA requirements that are appropriate for support of 10 CFR 830 Documented Safety Analyses (DSAs). However, many of the candidate accident analysis codes considered for toolbox status have uncertain SQA pedigree. Thus, before formally achieving the toolbox status, it will be necessary to define applicable SQA requirements or criteria, determine each computer code's baseline, and tailor an upgrade program for each code to meet the appropriate software quality assurance standards for safety-related applications.

The Implementation Plan contains commitment 4.2.1.2 to address this situation, stating "Establish SQA criteria for the safety analysis "toolbox" codes". The deliverable with this commitment is: SQA plan (including criteria) for toolbox codes.

This report supports completion of the commitment (4.2.1.2) by:

- Providing a plan for evaluating the SQA characteristics of the programs, procedures, and practices for the designated safety-related toolbox codes
- Identifying the criteria for evaluating the SQA adequacy of the DOE toolbox safety analysis computer codes. This document also addresses implementing procedures for meeting the primary criteria and assessing compliance.

1.1 Designated Safety Analysis Toolbox Codes

Safety analysis software for the DOE "toolbox" was designated by DOE/EH in March 2003 (DOE/EH, 2003). The supporting basis for this designation was provided by a DOE-chartered Safety Analysis Software Group in the technical report, *Selection of Computer Codes for DOE Safety Analysis Applications*, (August, 2002).[1] The codes for toolbox status, their version, and area of applicability are listed in Table 1-1. Later versions of the codes may be selected based on recommendations by the software developers and information obtained in the course of the SQA Implementation Program.

Eventually, each of these six codes and their respective development programs will undergo evaluation of their SQA attributes relative to established requirements identified in Task 4.2.1.2 and is termed a SQA evaluation. The SQA evaluation will assess those measures requiring action before the individual codes meet current SQA-compliant standards, and will be documented in a series of reports.

Table 1-1. Software Designated for DOE Safety Analysis Toolbox

Code	Version or Revision	Area of Applicability
ALOHA	5.2.3	Chemical Release/Dispersion and Consequence
CFAST	3.1.6	Fire Analysis
EPIcode	6.0	Chemical Release/Dispersion and Consequence
GENII	2.0*	Radiological Dispersion and Consequence
MACCS2**	1.12	Radiological Dispersion and Consequence
MELCOR	1.8.5	Leak Path Factor

* Version 1.485 may be advised for interim use before potential upgrades are completed. Recommendation to be based on near-term evaluation.
** Also MACCS, Version 1.5.11.1.

[1] Refer to internet-posted report available at https://www.hss.doe.gov/deprep/archive/rec/2002-1/NNSACCodes1.pdf.

1.2 Evaluation of Toolbox Codes

The quality assurance criteria identified in later sections of this report are defined as the set of established requirements, or basis, by which to evaluate each designated toolbox code. This evaluation process, a gap analysis, is commitment 4.2.1.3 in the IP:

> Perform a SQA evaluation to the toolbox codes to determine the actions needed to bring the codes into compliance with the SQA qualification criteria, and develop a schedule with milestones to upgrade each code based on the SQA evaluation results.

This process is a prerequisite step for software improvement. It will allow DOE to determine the current limitations and vulnerabilities of each code as well as help define and prioritize the steps required for improvement.

While it is required that each toolbox code owner provide full disclosure on the SQA programs, processes, and procedures used to develop their software, the gap analysis itself will be performed by an SQA evaluators. The SQA evaluator is independent of the code developer, but knowledgeable in the use of the software for accident analysis applications and current software development standards.

1.3 Uses of the Gap Analysis

The gap analysis will provide information to DOE, code developers, and code users.

DOE will see the following benefits:

- Estimate of the resources required to perform modifications to designated toolbox codes
- Basis for schedule and prioritization to upgrade each designated toolbox code.

Each code developer will be provided:

- Information on areas where software quality assurance improvements are needed to comply with industry SQA standards and practices
- Specific areas for improvement in terms of new versions of the software.

DOE safety analysts and code users will benefit from:

- Improved awareness of the strengths, limits, and vulnerable areas of each computer code
- Recommendations for code use in safety analysis application areas.

2.0 SQA Plan for Safety Analysis Toolbox Codes

A plan for conducting the review of the Software Quality Assurance programs, practices, and procedures for each of the six designated toolbox codes is required as part of the deliverables under IP commitment task 4.2.1.2. The participants in the review plan include the following:

SQA Evaluator - an independent reviewer of the computer software, who is not affiliated with the code developing organization. It is required this individual

- knows the SQA requirements at the level of rigor for accident analysis applications
- understands and has applied the software in question, and
- is aware of the overall context for the use of the software as part of the DOE accident analysis process.

Code Developer – the originator of the software, who is responsible for documenting SQA protocols associated with toolbox software, developing new versions of the subject software, addressing user questions, and resolving

DOE/EH – will develop the SQA Criteria and will be responsible for providing overall coordination and management of a Central Registry organization. The Central Registry will utilize the existing DOE Technical Standards Program procedures, processes, databases, and publications to provide long-term maintenance and control of designated "toolbox" codes. Through the Central Registry, DOE/EH will work closely with current "toolbox" code developers and user to preserve existing technical and programmatic responsibilities.

Each software developer will be requested to provide information on the programs and procedures associated with the development and maintenance of their software. A questionnaire for this purpose shall be transmitted to each software developer, requesting the following signed and approved documentation:

- Software Quality Assurance Plan
- Software Requirements Document
- Software Design Document
- Test Case Description and Report
- Software Configuration and Control Document
- Error Notification and Corrective Action Report
- User's Manual, and other relevant documentation (model description, weekly or monthly reports to code sponsor, etc.).

For documentation that is not available, the code developer will be requested to estimate the level of effort (LoE) required to complete the missing reports. The code developer will also be asked to assess the adequacy of the existing set of documents, and estimate resource requirements to upgrade any of those that are deemed deficient. If software errors or other deficiencies in the model are known, the code developer will estimate the schedule for resolving these issues, or specify the resources needed to perform the corrective actions.

The SQA Evaluator will then perform and document a review of the software, using the inputs from the code developer, including documentation, resource estimates, and other communications. In cases where the software developer is unable to supply inputs to the SQA Evaluator, the gap analysis will proceed with alternative sources of information. Examples of alternative information are previous reviews, older documentation from the code developer, technical and journal articles, and previous software comparison studies.

Additional detail on the use of the criteria in the SQA evaluation is discussed in Section 3.

2.1 Software Categorization and Classification

Three DOE site programs on SQA were surveyed for applicability as bases for categorizing and classifying the designated toolbox software. Included were the programs of Sandia National Laboratories (SNL), the Yucca Mountain Project (YMP) and the Savannah River Site (SRS) (SNL, 2003; OCRWM 2003a-2003c;WSRC, 2003, 2003a-2003c). The present section applies elements drawn from these three SQA programs in recommending both a plan and set of evaluation criteria to evaluate the safety analysis software proposed for the DOE toolbox.

The initial step in the overall process of establishing a plan for software evaluation and the evaluation criteria is to identify the category of the software. The category of the safety analysis software may be thought of as the software engineering approach, with the following types:

1. Acquired or purchased software
2. New software, in development
3. Existing software is that is not described by (1) or (2), and that is wholly or partially non-compliant with software criteria (discussed in Section 3), or
4. A combination of the above.

The first category is typically represented by process control software for controlling and monitoring a SSC in a nuclear facility. The second category of software includes new software specification, design, and implementation, where SQA compliance tasks are performed as the new software is built. The third category, existing software, seems to best characterize the six designated safety analysis codes (DOE, 2002b). While most of the designated software references one or more of the primary SQA standards in its respective documentation, not one is fully compliant.

The primary SQA criteria, discussed in the following section, recommend that the level of SQA associated with a computer code be commensurate with the importance of the software application. Thus, the second determination to be made prior to formally beginning the evaluation is software classification. The classification of the software level for a specific computer code is a determination of the importance of the software and its intended use for a given application. The classification and the category of the software drive the requirements that must be satisfied, and is based on graded application considerations. Table 2-1 provides a hierarchy for software classification using a five-tier system, containing Levels A through E.

In terms of application to safety analysis software, both level A and B classifications imply that the most stringent requirements must be met. However, Level A is reserved for software *whose output is used in the operation of a SSC with no additional evaluation or review prior to taking action*. This intent is in contrast to Level B software *whose output is used in the operation of a SSC, but is subject to evaluation or review prior to taking action*.
Recognizing that the designated toolbox software is used in applications where the output of the software is part of the evaluation in accident analysis, and is typically subject to thorough technical review, the most applicable classification for the designated

Table 2-1. Software Level Hierarchy by Safety Analysis Application[2]

Level	Classification Basis
A	• Software applications that have a direct effect on nuclear safety protection systems that keep exposure to the general public below the off-site regulatory or evaluation guidelines.
B	• Software applications whose failure to properly function may have an indirect effect on nuclear safety protection systems or toxic materials hazard systems that are used to keep nuclear or toxic material hazard exposure to the general public and workers below regulatory or evaluation guidelines. • Software applications whose results are used to make decisions that could result in death or serious injury or are part of the evaluation in accident analyses.
C	• Software applications whose failure to perform as expected would not affect nuclear safety but would have an unacceptable impact by causing loss of: - greater than $2 million dollars production investment value and/or recovery cost - primary program capabilities in excess of six months. • Software applications important to continued operations of the business and that which is used to support decisions regarding operating activities. • Software applications used to comply with regulatory laws, environmental permits or regulations and/or commitments to compliance. • Software applications required by the site/laboratory Emergency Plan for environmental monitoring or for communications with Local, State and Federal Government agencies.
D	Software applications important to the day-to-day administration of the business but whose failure to perform as intended will not adversely affect the safety or reliability of operations or will not result in losses exceeding $2 Million Dollars or result in a six month loss of program capabilities.
E	Software that is within scope of this application but does not meet the criteria specified in the above classification levels.

[2] Based on WSRC (2003).

safety analysis toolbox software is Level B, i.e.,

- *Software applications whose failure to properly function may have an indirect effect on nuclear safety protection systems or toxic materials hazard systems that are used to keep nuclear or toxic material hazard exposure to the general public and workers below regulatory or evaluation guidelines, or*
- *Software applications whose results are used to make decisions that could result in death or serious injury or are part of the evaluation in accident analyses.*

The Level B classification reflects the understanding of the overall context for use of the toolbox codes, i.e., supporting development of 10 CFR 830 compliant DSAs, because these computer codes are not used as the sole basis for making safety-related decisions.

Table A-1 in Appendix A cross-references software classification with the functional classification process. Other safety-related software, including safety system, process control, and design software can be assessed using this, or similar, classification-based hierarchy. More detail on the software classification for the accident analysis codes is found in Appendix A.

2.2 Application of Implementation Procedures for Existing Software

Defining the designated toolbox software under the Level B classification structures the requirements and responsibilities for the quality assurance of the computer software. As the primary SQA criteria, ASME NQA-1-2000, in Section 302, outlines requirements for ". . . *existing software* not previously approved under a program consistent with this Standard". Sandia National Laboratories, Yucca Mountain Project and the Savannah River Site have proceduralized these requirements in more detail in their site software implementation procedures. Although the detail varies from site to site, each set of procedures prescribes the steps to be taken to classify, evaluate, validate, place under configuration control, and control software of this category in accordance with the life cycle requirements.

In brief, an evaluation of existing software for accident analysis applications:

a. determines the adequacy of the subject software documentation to support testing, operation, and maintenance.
b. identifies activities to be performed throughout the applicable lifecycle of the software including preparation of required documentation and performance of required reviews and/or tests,
c. determines the software's capabilities and limitations for intended use,
d. specifies test plans and test cases required to validate the capabilities within the stated limitations,
e. identifies instructions for software use within the limits of its capabilities,
f. identifies any exceptions to the lifecycle documentation and its justification.

As noted earlier, the SQA Evaluator obtains the appropriate input documentation from the code developer, or performs a review of the documentation from the code developer and/or other

alternative sources. An input template questionnaire shall be sent to the software developers to expedite transmittal of the required information to the SQA evaluators.

Table 2-2 contains a plan for evaluating existing software, developed outside of the full requirements of the primary criteria (outlined in the next section), and defined as Level B software for safety analysis applications. This procedure, to be followed by the SQA evaluator, provides instructions for the evaluation of existing safety analysis software, referencing the detailed procedures and criteria discussed in Section 3 of this report. Table 2-2 is based on an evaluation procedure used at SRS for existing software, edited for applicability for safety analysis software (WSRC, 2003c).

Phases 6 and 7 on Software Training and Engineering Planning (Upgrades), respectively, have been added to the overall evaluation process.

Table 2-2. – Plan for SQA Evaluation of Existing Safety Analysis Software

Phase	Procedure
1. Prerequisites	a. Determine that sufficient information is provided by the software developer to allow it to be properly classified for its intended end-use. b. Review SQAP per applicable requirements in Table 3-3.
2. Software Engineering Process Requirements	a. Review 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 SQAP, e.g., • Software Requirements Document • Software Design Document • Test Case Description and Report • Software Configuration and Control Document • Error Notification and Corrective Action Report, and • User's Instructions (alternatively, a User's Manual), Model Description (if this information has not already been covered). c. Identify documents that are acceptable from SQA perspective. Note inadequate documents as appropriate.
3. Software Product Technical/ Functional Requirements	a. Review requirements documentation to determine if requirements support intended use in Safety Analysis. Document this determination in gap analysis document. b. Review previously conducted software testing to verify that it sufficiently demonstrated software performance required by the Software Requirements Document. Document this determination in the gap analysis document.
4. Testing	a. Determine whether past software testing for the software being evaluated provides adequate assurance that software product/technical requirements have been met. Obtain documentation of this determination. Document this determination in the gap analysis report. b. (Optional) Recommend test plans/cases/acceptance criteria as needed per the SQAP if testing not performed or incomplete.
5. New Software Baseline	a. Recommend remedial actions for upgrading software documents that constitute baseline for software. Recommendations can include complete revision or providing new documentation. A complete list of baseline documents includes: • Software Quality Assurance Plan • Software Requirements Document • Software Design Document • Test Case Description and Report • Software Configuration and Control • Error Notification and Corrective Action Report, and • User's Instructions (alternatively, a User's Manual) b. Provide recommendation for central registry as to minimum set of SQA documents to constitute new baseline per the SQAP.

Table 2-2. – Plan for SQA Evaluation of Existing Safety Analysis Software (continued)

Phase	Procedure
6. Training	a. Identify current training programs provided by developer. b. Determine applicability of training for DOE facility safety analysis.
7. Software Engineering Planning	a. Identify planned improvements of software to comply with SQA requirements. b. Determine software modifications planned by developer. c. Provide recommendations from user community. d. Estimate resources required to upgrade software.

3.0 Software Quality Assurance Criteria for Safety Analysis Toolbox Codes

This section provides a rationale for selection the primary set of primary SQA criteria for use in assessing the designated toolbox computer software.

The primary criteria are those in the Quality Assurance rule, Subpart A to 10 CFR 830. Subpart A establishes quality assurance requirements for DOE contractors conducting activities including providing items or services, that affect, or may affect, the nuclear safety of DOE nuclear facilities.

While several national and international sets of software quality assurance partially meet the needs of assuring software quality in the nuclear sector and provide guidance to following the Quality Assurance rule, it is concluded that the ASME NQA-1 requirements best address safety analysis software and cover the full spectrum of needs for this type of software. NQA-1 is nuclear industry's response to 10 CFR 50, Appendix B (DNFSB, 2001), and is referenced in 10 CFR 830 Subpart A. It provides guidance for complying with Nuclear Safety requirements. It incorporates the basic criteria from 10 CFR 50, Appendix B, 10 CFR 830 Subpart A and references key criteria from Institute of Electronics and Electrical Engineers (IEEE) standards. An organizational structure and assignment of responsibilities is formally prescribed in NQA-1 such that

- management establishes overall expectations for effective QA program implementation and is ultimately responsible for the end result;
- quality is achieved and maintained by those performing work; and
- quality is verified by those not directly responsible for performing the work.

In other words, there are clear, unambiguous roles delineated in NQA-1, and defined independence in performing various phases of work. The functional roles and independence characteristics are prerequisites to developing and maintaining a controlled approach to developing sound safety analysis software.

Both NQA-1a-1999 and NQA-1-2000 emphasize performance-based practices and graded application, yet reduce prescriptive requirements and redundancy (ASME, 2002). Thus, the NQA-1 standard is intended by its authors to be applied in a graded approach manner. This intent of the NQA Committee is clear from the recommendation of judicious application of the entire standard or portions of the standard. The standard goes on to indicate

> The extent to which this Standard should be applied will depend upon the specific type of nuclear facility, items, or services involved and the nature and scope and the relative importance of the activities being performed. The extent of application is to be determined by the organization imposing the Standard (ASME, 2000).

A major theme to changes in NQA-1 has been protecting the health and safety of the public while performing work that meets requirements. This goal is in line with nuclear safety directives and guidance from the Department of Energy, including DOE-STD-3009-94 and other "safe harbor" methodologies listed in Table 2 in Subpart B to 10 CFR 830. While many of the

requirements from ISO 9001 (or ISO 9000-3) can be considered to complement NQA-1, the fundamental intent of ISO 9001 is as a quality management standard. Moreover, it is not specifically directed at the health and safety concerns.

Finally, it should be noted that many contractors have already based their respective site software quality assurance programs on some version of NQA-1. To shift to another system for benchmarking safety analysis codes would demand high resource commitments without a commensurate increase in the level of software quality achieved.

In summary, 10 CFR 830 Subpart A, and the NQA-1-2000, Subpart 2.7 and related Part I requirements, primarily Requirements 3 (*Design Control*/Section 800 *Software Design Control*) and 11 (*Test Control*/Section 400 *Computer Program Test Procedures*), are recommended as the primary set of SQA criteria for the evaluation of safety-related computer software. This selection is based on

- Nuclear industry precedent with ASME NQA standards
- Federal and commercial sectors continued involvement with, and maintenance of the ASME NQA standards
- Quality assurance perspective through connection with 10 CFR 50 Appendix B and 10 CFR 70
- Independence of roles in developing and maintaining software, among management, work performers, and work reviewers
- Graded application based on safety, risk, and hazard analysis of the function of the software
- Focus on protection of the public and workers
- Long-standing presence and incorporation with many DOE contractors' quality assurance programs, with focus on nuclear safety, and
- Completeness and relevance to scientific, applied research, design, analysis and nuclear engineering software.

Other Sections of NQA-1 and applicable DOE Directives will form a secondary set of criteria. These include, but are not limited to, DOE O 414.1A and DOE N 411.1. Appendix B summarizes key DOE Orders and directives, and related standards pertinent to quality assurance and SQA. These documents were reviewed prior to identifying the primary SQA criteria.

Appendix C provides a comparison of sections from NQA-1-1997, Part II, Section 2.7 (Quality Assurance Requirements for Computer Software for Nuclear Facility Applications), with the 1999 and 2000 versions of the standard. In general, Subpart 2.7 has been updated in its bases from ANSI/IEEE 729, *Glossary of Software Engineering Terminology* and ANSI/IEEE 1012, *Software Verification and Validation Plans*, to IEEE Std. 7-4.3.2-1993, *IEEE Standard Criteria for Digital Computers in safety Systems of Nuclear Power Generating Stations* and ANSI/IEEE Std. 610.12-1990, *Glossary of Software Engineering Terminology*. Consistent with the change in IEEE standards as a basis, the Subpart also uses software design verification and testing. Section 3.2 summarizes key sections from NQA-1-2000, Part II, and Section 2.7.

3.1 10 CFR 830 Subpart A

Subpart A to 10 CFR 830 establishes quality assurance requirements for DOE contractors conducting activities including providing items or services, that affect, or may affect, the nuclear safety of DOE nuclear facilities. Section 830.121 describes a requisite quality assurance program (QAP) its applicability, frequency of updates, and directs the contractor to describe how criteria (Section 830.122) are met. It also specifies integration with the Safety Management System and recommends use of voluntary consensus standards.

Ten broad quality assurance criteria are described in Section 830.122. Each quality assurance criterion is stated as a performance expectation without specification of the methods for achieving the desired result. Instead, contractors are directed to national and international standards to develop effective and efficient QAPs. The management, performance, and assessment criteria include:

1 – Management Program
2 - Management/Personnel Training and Qualification
3 - Management/Quality Improvement
4 – Management/Documents and Records
5 – Performance/Work Processes
6 – Performance/Design
7 – Performance/Procurement
8 – Performance/Inspection and Acceptance Testing
9 – Assessment/Management Assessment
10 – Assessment/Independent Assessment.

The DOE implementation guide for quality assurance requirements from the 10 CFR 830 rule is DOE G 414.1-2. DOE G 414.1-2 includes a discussion of standards use, and references the most widely accepted standards for quality assurance.

3.2 NQA-1-2000, Part II, Section 2.7 (Quality Assurance Requirements for Computer Software for Nuclear Facility Applications)

The core set of requirements for quality assurance of safety analysis software is contained in Subpart 2.7 of ASME NQA-1-2000. Subpart 2.7 provides requirements for the acquisition, development, operation, maintenance, and retirement of software. However, implementation of these requirements by a code developer should follow a prescriptive set of instructions. Subpart 2.7 notes that "The appropriate requirements of this Subpart shall be implemented through the policies, procedures, plans, specifications, or work practices, etc., that provide the framework for software engineering activities". Thus, it is expected that the safety analysis software owners/vendors have used a documented procedural basis to develop their respective software. Furthermore, it would be expected that the procedures meet NQA-1 requirements, or those of an equivalent basis.

Four broad elements are included in the scope of software engineering activities described in Subpart 2.7:

(a) software acquisition methods for controlling the acquisition process for software and software services;
(b) software engineering method(s) used to manage the software life-cycle activities;
(c) application of standards, conventions, and other work practices that support the software life cycle;
(d) controls for support software used to develop, operate, and maintain computer programs.

Section 200 covers General Requirements, including Documentation, Review, Configuration Management, and Problem Reporting and Corrective Action. Section 300 outlines software requirements according to the type of acquisition. Section 400 contains requirements on documentation, and the planning and performance of software life cycle activities. Included are Software Design Requirements, Software Design, Implementation, Acceptance Testing, Operation, Maintenance, and Retirement.

Other requirements of NQA-1, specifically sections from Part I, are referenced in the body of Subpart 2.7 or are described as recommended practices, and should be referenced as appropriate (Table 3-1). Part IV, Subpart 4.1 is an application guide with a discussion of the requirements and how those requirements may apply in various situations where software is used. These supporting sections to Subpart 2.7 are typically cited as "applicable parts".

Table 3-1. Applicable Sections from NQA-1 Supporting Software Development and Maintenance

Part	Requirement	Section
I	2 – Quality Assurance Program	100 – Basic 200 – Indoctrination and Training
I	3 – Design Control	400 – Design Analysis 800 – Software Design Control
I	4 – Procurement Document Control	Applicable Requirements to Software
I	7 – Control of Purchased Items and Services	Applicable Requirements to Software
I	11 – Test Control	100 – Basic 200 – Test Requirements 400 – Computer Program Test Procedures 500 – Test Results 600 – Test Records
IV	4.1 Application Appendix – Guide on Quality Assurance Requirements for Software	100 – General 200 – General Requirements 300- Software Acquisition 400 – Software Engineering Method 500 – Standards, Conventions, and Other Work Practices 600 – Support Software 601 – Software Tools 602 – System Software

3.3 Department of Energy Order 414.1A

DOE O 414.1A describes how to establish an effective management system, in terms of quality assurance programs or QAPs, using the performance requirements identified in 10 CFR 830.122 and reiterated in DOE O 414.1A. Coupled with technical consensus standards where appropriate, DOE O 414.1A will ensure:

- senior management provides planning, organization, direction, control, and support to achieve DOE objectives,
- line organization functions and responsibilities are defined, and
- each DOE element reviews, evaluates and improves its overall performance using a rigorous assessment process.

The Order states that DOE elements (including NNSA) must implement the quality assurance criteria in a manner sufficient to achieve adequate protection of workers, public and the environment, taking into account the work to be performed and the associated hazards. Quality assurance criteria in 10 CFR 830.122 must be used to develop the QAPs, with the latter describing how the criteria are satisfied. Use of the graded approach must be described.

3.4 Implementation of Primary SQA Criteria

The primary criteria require a context and implementation procedures before a computer code can be assessed in terms of its compliance. This section provides a conceptual approach for evaluating the SQA of safety-related software by first noting the rule and supporting DOE directives in quality assurance. A set of implementing procedures is then proposed, based on a limited survey of SQA practices in the DOE Complex, as the applicable "working level" set of instructions for complying with the primary SQA criteria.

3.4.1 Conceptual Approach for SQA Evaluation

The proposed framework for SQA evaluation of the safety analysis codes applies 10 CFR 830 Subpart A Quality Assurance and primary SQA criteria (NQA-1a 1999 and NQA-1 2000) supplemented, and is shown in a schematic in Figure 3-1. The Nuclear Safety Management rule, 10 CFR 830, Subpart A on Quality Assurance provides the over-arching context for use of the primary SQA criteria. Specific DOE directives are factored into the SQA evaluation process, including

- DOE O 414.1A (Quality Assurance)
- Other QA Program Standards, QC-1, DOE-RW-0333P, etc.[3]
- DOE Notice 411.1
- DOE Standards and Implementation Guides.

[3] Refer to Appendix B for summaries of related DOE directives.

The SQA evaluation process will review the development as well as status of each of the toolbox codes considering the contractual agreement between code sponsor and the code developer (central, gray box in Figure 3-1, "Code Developer – Sponsor Contractual Requirements").[4] However, one SQA implementing procedural basis ("Code Developer Software Engineering and Maintenance Procedures") will be applied to assess how well each code complies with the primary SQA criteria. Consensus standards from IEEE, and listed in Appendix C will be used as needed to support the procedural basis.

[4] Review of certain software information is subject to availability from the code developer.

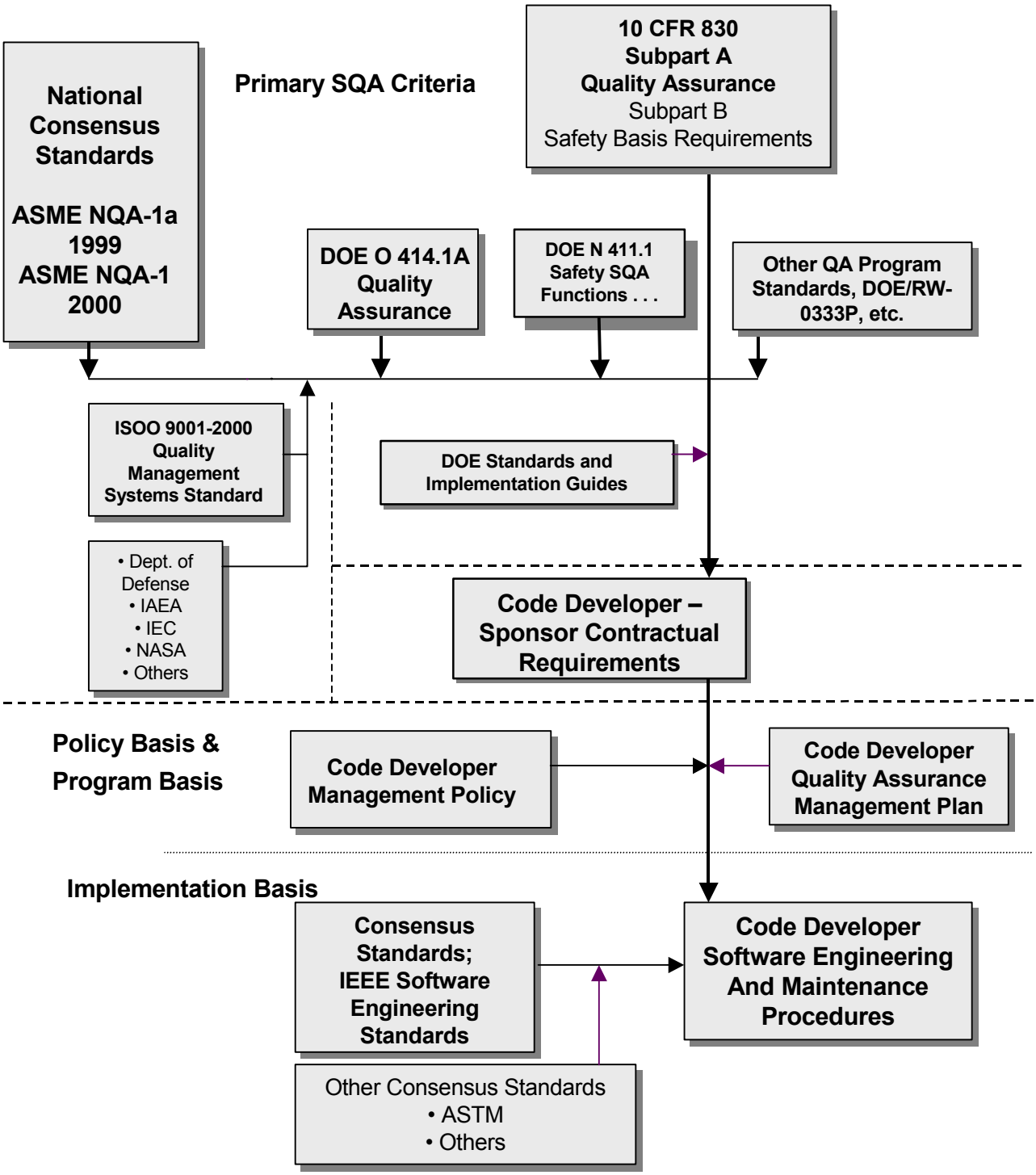

Figure 3-1. SQA Program Flowdown - Safety Analysis Software

Information supplied from each of the designated toolbox code developers will be used to guide the evaluation process. An information template will be developed to facilitate the assessment and will be based on the selected procedural basis discussed below.

3.4.2 Selection of Implementing Procedures

As noted previously, requirements from NQA-1-2000 are not met directly, but require implementing procedures with sufficient detail to guide appropriate actions for each computer code. The implementing procedures for meeting NQA-1-level requirements from Sandia National Laboratories, the Savannah River Site, and the Yucca Mountain Project were reviewed for application from the perspective of the computer software developer. The final procedural basis shown here is a merged set composed of procedures from these sources. However, it is based extensively on procedures from the Savannah River Site.

Consequently, the requirements matrix in Table 3-2 developed for Classification Level B software are can be applied and is based on the origin of the software.[5] In other words, the requirements differ, depending on whether *the software is under development, exists but did not follow NQA-1-2000 or similar primary criteria,* or *is being purchased*. Review of the six safety analysis codes designated for the toolbox suggests that this software is in "Existing" category, and the specific requirements listed in Table 3-2 under "Level B Existing" will be applicable. The Table 3-2 indication of meeting the requirement specifies it is "required" or is "graded". In this application, "required" steps must be performed, while "graded" steps are those commensurate with the safety analysis application, and are left to the judgment of the SQA evaluator. Several changes have been made to the matrix in terms of required versus graded to better enable evaluation of the software developer's programs, processes, and procedures.

[5] See full WSRC (2003) for full text. A summary is listed here.

Table 3-2. Software Requirements Matrix for Level B Classification Application

REQUIREMENTS	Computer Software Origin		
	Level B Development	Level B Existing	Level B Purchased
1. Software Classification	Required*	Required	Required
2. SQA Procedures/Plans	Required	Required	Required
3. Dedication	Graded**	Graded	Required
4. Evaluation	Graded	Required	Graded
5. Requirements	Required	Required	Required
6. Design	Required	Required	Graded
7. Implementation	Required	Required	Graded
8. Testing	Required	Required	Required
9. User Instructions	Required	Required	Required
10. Acceptance Test	Required	Required	Required
11. Operation & Maintenance	Required	Required	Required
12. Configuration Control	Required	Required	Required
13. Error Impact	Graded	Graded	Graded
14. Access Control	Required	Required	Required

* Required for the computer software; ** Graded depending on the application, and based on judgment of SQA Evaluator.

The full implementing procedures for demonstrating compliance with the fourteen NQA-1-2000 fourteen requirements are shown in Table 3-3. Table 3-3 provides detailed criteria for each applicable requirement, and the matching ASME NQA-1-2000 section and consensus standard(s).[6] The software developer would need to demonstrate that the program, procedures, and practices followed are compliant with the requirements listed here.

Files, reports, telephone conferences, and other documented communications can provide indication that actions have been performed in a SQA program, and these can be reported by the developer to the SQA evaluator. However, several formal documents explicitly demonstrate compliance with the primary criteria. The following documents are examples of this compliance class and are specifically noted in Table 3-3:

- Software Quality Assurance Plan
- Software Requirements Document
- Software Design Document
- Test Case Description and Report
- Software Configuration and Control Document
- Error Notification and Corrective Action Report, and
- User's Instructions (alternatively, a user's manual).

[6] Based on QAP 20-1 (WSRC, 2003) and implementing procedures (WSRC, 2003a).

While the procedures listed in Table 3-3 are complete for evaluating all categories of software, a subset will be treated by the SQA evaluation process for the existing safety analysis software designated for the DOE Toolbox. The requirements to be evaluated, as labeled in Table 3-3, are:

Table 3-3 No.	Requirement
1	Software Classification
2	SQA Procedures/Plans
5	Requirements Phase
6	Design Phase
7	Implementation Phase
8	Testing Phase
9	User Instructions
10	Acceptance Test
12	Configuration Control
13	Error Notification.

Requirements 3 (Dedication), 11 (Operation and Maintenance), and 14 (Access Control), are not applicable for the software development process, and thus are not evaluated in this review. Requirement 4 (Evaluation) is an outline of the minimum steps to be undertaken in a software review, and is complied with by performing the steps listed above in the full evaluation process.

Table 3-3.1 Software Documentation Requirements Matrix for Level B Classification Application – Classification Through Evaluation

	REQUIREMENT	PROCEDURE	ASME NQA-1 2000 Section / Consensus Standards
1.	**Software Classification (Note 1.)**	Determine whether the code developer has provided sufficient information to allow the user to make an informed decision on the classification level of the software, i.e., its designation is based on: its direct effect on nuclear safety protection systems; indirect effect on nuclear safety protection systems or toxic material hazard systems.	ASME NQA-1 2000 Section 200;
2.	**SQA Procedures/Plans**	Verify that procedures/plans for SQA (**SQA Plan**) have been prepared which identify the following based on the nature, complexity, and intended use(s) of the subject software • Identify organizations responsible for performing work; independent reviews, etc. • Software engineering methods • Documentation to be required as part of program • Standards, conventions, techniques, and/or methodologies which shall be used to guide the software development, methods to ensure compliance with the same • Software reviews and schedule. • Methods for error reporting and corrective actions.	ASME NQA-1 2000 Section 200; IEEE Std. 730, *IEEE Standard for Software Quality Assurance Plans*
3.	**Dedication**	Applicable to user organizations planning to purchase commercial off-the-shelf software.	ASME NQA-1 2000 Section 300; EPRI NP-5652
4.	**Evaluation**	This procedure is a recommended course of action, i.e., set of steps for evaluating the existing category of software. It is a summary statement of the minimum required steps. Safety analysis software, not developed in accordance with the procedures compliant with the primary SQA criteria, shall be classified, evaluated, validated, placed under configuration control, and controlled in accordance with the life cycle requirements for Level B classification software. The following steps are advised: a. determine the adequacy of software documentation to support testing, operation, and maintenance. b. identify activities to be performed throughout the applicable life cycle of the software including preparation of required documentation and performance of required reviews and/or tests. c. determine the software's capabilities and limitations for intended use d. specify test plans and test cases required to validate the capabilities within the stated limitations e. identify instructions for software use within the limits of its capabilities f. identify any exceptions to the life cycle documentation and its justification. The SQA evaluator is to obtain documentation from the software developer to enable the evaluation to proceed.	ASME NQA-1 2000 Section 302

Note: 1. The user organization will typically determine Classification Level for Software. Detailed implementation requirements for application of the software classification process should be provided in user organization's manuals.

Table 3-3.2 Software Documentation Requirements Matrix for Level B Classification Application – Requirements

	REQUIREMENT	PROCEDURE	ASME NQA-1 2000 Section / Consensus Standards
5.	**Requirements Phase**	Verify that software requirements for the subject software have been established. Specify, document, review, and approve the requirements for the subject software that must be satisfied. The requirements shall define the functions to be performed by the software, and shall provide detail and information necessary to design software. A **Software Requirements Document**, or equivalent, should define requirements for, functionality, performance, design inputs, design constraints, installation considerations, operating systems (if applicable), and external interfaces necessary to design the software. Acceptance criteria shall be established in the software requirements documentation for each of the identified requirements. Such criteria shall be used for verification/validation planning and performance as defined in each related life cycle phase.	ASME NQA-1 2000 Section 401; IEEE Standard 830, *Software Requirements Specifications*

Table 3-3.3 Software Documentation Requirements Matrix for Level B Classification Application – Design Phase

	REQUIREMENT	PROCEDURE	ASME NQA-1 2000 Section / Consensus Standards
6.	**Design Phase**	Verify a software design was developed, documented, and reviewed and controlled. The code developer should have prescribed and documented the design activities to the level of detail necessary to permit the design process to be carried out and to permit verification that the design met requirements. The following design elements should be present and documented: a. The design should specify the interfaces, overall structure (control and data flow) and the reduction of the overall structure into physical solutions (algorithms, equations, control logic, and data structures). b Computer programs were designed as an integral part of an overall system. Therefore, evidence should be present that the software design considered the computer program's operating environment. c. Evidence of measures to mitigate the consequences of problems should be an integral part of software design. These potential problems include external and internal abnormal conditions and events that can affect the computer program. A **Software Design Document**, or the equivalent, should be available, and should contain: - a description of the major components of the software design as they relate to the software requirements; - a technical description of the software with respect to the theoretical basis, mathematical model, control flow, data flow, control logic, data structure, numerical methods, physical models, process flow, process structures, and applicable relationship between data structure and process standards; - a description of the allowable or prescribed ranges for inputs and outputs; - the design described in a manner that can be translated into code; and - a description of the approach to be taken for intended test activities based on the requirements and design that specify the hardware and software configuration to be used during test execution. Review and approval: The organization responsible for the design should have identified and documented the particular verification methods to be used and assured that an Independent Review was performed and documented. This review should have evaluated the technical adequacy of the design approach; assure internal completeness, consistency, clarity, and correctness of the software design; and verified that the software design is traceable to the requirements. The organization responsible for the design should have assured that the test results adequately demonstrated the requirements were met.	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*

Table 3-3.4 Software Documentation Requirements Matrix for Level B Classification Application – Design & Implementation

	REQUIREMENT	PROCEDURE	ASME NQA-1 2000 Section / Consensus Standards
6.	**Design Phase (continued)**	The Independent Review shall be performed by competent individual(s) other than those who developed and documented the original design, but who may be from the same organization. The results of the Independent Review shall be documented with the identification of the verifier indicated. When review alone is not adequate to determine if requirements are met, alternate calculations shall be used, or tests shall be developed and integrated into the appropriate activities of the software development cycle. Software design documentation shall be completed prior to finalizing the Independent Review. The extent of the IR and the methods chosen should be shown to be a function of: (a) the importance to safety, (b) the complexity of the software, (c) the degree of standardization, and (d) the similarity with previously proven software.	
7.	**Implementation Phase**	Verify that there is evidence of the implementation process resulting in software products such as computer program listings and instructions for computer program use. There should be evidence that implemented software was analyzed to identify and correct errors. The source code finalized at this time should have been placed under configuration control. Documentation for this phase shall include a copy of the software, test case description and associated criteria that are traceable to the software requirements and design documentation.	ASME NQA-1 2000 Section 204; IEEE Standard 1016.1, *IEEE Guide for Software Design Descriptions;* IEEE Standard 1016-1998, *IEEE Recommended Practice for Software Design Descriptions*

Table 3-3.5 Software Documentation Requirements Matrix for Level B Classification Application – Testing Phase

	REQUIREMENT	PROCEDURE	ASME NQA-1 2000 Section / Consensus Standards
8.	**Testing Phase**	During this phase the software shall be validated by executing the test cases. Failure to successfully execute the test cases shall be reviewed to determine if modification of the requirements, the design, the implementation, or the test plans and test cases are required. Testing shall demonstrate the capability of the software to produce valid results for test cases encompassing the range of permitted usage defined by the program documentation. Such activities shall ensure that the software adequately and correctly performs all intended functions. Testing for safety analysis software should have demonstrated, as appropriate, that the computer program: (1) properly handles abnormal conditions and events as well as credible failures (2) does not perform adverse unintended functions. Test Phase activities shall consist of the testing of the software to assure adherence to requirements, and to assure that the software produces correct results for the test cases specified. Acceptable methods for evaluating the adequacy of the software test case results include: (1) analysis without computer assistance (2) other validated computer program(s), (3) experiments and tests, (4) standard problems with known solutions, (5) confirmed published data and correlations. **Test Phase** documentation should include test procedures or plans and the results of the execution of test cases. The test results documentation should demonstrate successful completion of all test cases or the resolution of unsuccessful test cases and provide direct traceability between the test results and specified software requirements. Test procedures or plans shall specify the following, <u>as applicable</u>: (1) required tests and test sequence, (2) required range of input parameters, (3) identification of the stages at which testing is required, (4) requirements for testing logic branches, (5) requirements for hardware integration, (6) anticipated output values, (7) acceptance criteria, (8) reports, records, standard formatting, and conventions, (9) identification of operating environment, support software, software tools or system software, Hardware Operating System(s) and/or limitations.	ASME NQA-1 2000 Section 404; IEEE Std. 829, *IEEE Standard for Software Test Documentation;* IEEE Standard 1008, *Software Unit Testing*

Table 3-3.6 Software Documentation Requirements Matrix for Level B Classification Application – User Instructions and Acceptance Test

	REQUIREMENT	PROCEDURE	ASME NQA-1 2000 Section / Consensus Standards
9.	**User Instructions (Note 2)**	User instructions (**User's Manual or Guide**) shall include: a. approved operating systems (for cases where source code is provided, applicable compilers should be noted) b. description of the user's interaction with the software, c. a description of any required training necessary to use the software, d. input and output specifications, e. input and output formats, f. a description of software and hardware limitations, g. a description of user messages initiated as a result of improper input and how the user can respond, h. information for obtaining user and maintenance support.	ASME NQA-1 2000 Section 203; IEEE Standard 1063, *IEEE Standard for Software User Documentation*
10.	**Acceptance Test (Note 3)**	During this phase the software becomes part of a system incorporating applicable software components, hardware, and data and is accepted for use. For the installation and acceptance phase: (1) Acceptance testing shall include a comprehensive test in the operating environment. (2) Acceptance testing shall be performed prior to approval of the computer program for use. (3) Software validation shall be performed to ensure that the installed software product satisfies the specified software requirements. The engineering function (i.e., an engineering operation an item is required to perform to meet the component or system design basis) shall determine the acceptance testing to be performed prior to approval of the computer program for use. Installation and acceptance phase documentation shall include results of the execution of test cases for system installation and integration, user instructions (Refer to Requirement 9 above), and documentation of the acceptance of the software for operational use.	ASME NQA-1 2000 Section 404; IEEE Std. 829, *IEEE Standard for Software Test Documentation;* IEEE Standard 1008, *Software Unit Testing*

Note 2. Either in this requirement or in an earlier one, a description of the model should be documented.
Note 3. The user organization will usually perform steps 1 through 3, but the developing organization may apply applicable parts of Requirement 10.

Table 3-3.7 Software Documentation Requirements Matrix for Level B Classification Application – Operation and Maintenance and Configuration Control

	REQUIREMENT	PROCEDURE	ASME NQA-1 2000 Section / Consensus Standards
11.	**Operation and Maintenance (Note 4.)**	During this phase, software shall be controlled to remove latent errors (corrective maintenance), to respond to new or revised requirements (enhancement), or to adapt the software to changes in the operating environment (adaptive maintenance). Software modifications shall be approved, documented, verified and validated, and controlled in accordance with the related life cycle phases. The validation of modifications shall be subject to selective regression testing to detect errors introduced during the modification of software or operating system components to verify that the modifications have not caused unintended adverse effects and to verify that the modified software still meets its specified requirements. Test cases shall be developed and documented to permit confirmation of acceptable performance of the software in the environment in which the software is used. Test cases shall be run whenever the software is installed on a different computer, or when significant hardware or operating system configuration changes are made. Periodic in-use manual or automatic self-check in-use tests shall be prescribed and performed for those computer programs where computer program errors, data errors, computer hardware failures, or instrument drift can affect required performance.	ASME NQA-1 2000 Section 405; ASME NQA-1 2000 Section 406;
12.	**Configuration Control**	The methods to be used to control, uniquely identify, describe, and document the configuration of each version or update of a computer program (for example, source, object, back-up files) and its related documentation (for example, software design requirements, instructions for computer program use, test plans, and results) shall be described in implementing procedures. Such procedures shall meet applicable criteria for configuration identification, change control and configuration status accounting.	ASME NQA-1 2000 Section 203;

Note 4. Not all procedures are applicable to toolbox code category of software.

Table 3-3.8 Software Documentation Requirements Matrix for Level B Classification Application – Error Impact and Access Control

	REQUIREMENTS	PROCEDURE	ASME NQA-1 2000 Section / Consensus Standards
13.	**Error Impact**	The problem reporting and corrective action process used by the developing organization should address the appropriate requirements of the site/laboratory corrective action system and the following elements: a. Method(s) for documenting (**Error Notification and Corrective Action Report**), evaluating, and correcting software problems shall: (1) describe the evaluation process for determining whether a reported problem is an error; and (2) define the responsibilities for disposition of the problem reports, including notification to the originator of the results of the evaluation. b. When the problem is determined to be an error, the method shall provide, as appropriate, for: (1) how the error relates to appropriate software engineering elements; (2) how the error impacts past and present use of the computer program; (3) how the corrective action impacts previous development activities; (4) how the users are notified of the identified error, its impact; and how to avoid the error, pending implementation of corrective actions. The methods to be used for resolving software problems and taking appropriate action shall be described in implementing procedures that comply with site/laboratory procedures.	ASME NQA-1 2000 Section 204; IEEE Standard 1063, *IEEE Standard for Software User Documentation*
14.	**Access Control**	N/A – This requirement would typically be handled by the user organization.	ASME NQA-1 2000 Section 405

4.0 Overall Process for SQA Evaluation of Safety Analysis Toolbox Codes

The overall process for the SQA evaluation of each safety analysis toolbox code is summarized in Table 4-1. Table 4-1 lists actions, and recommended responsible organizations.

Most actions in Table 4-1 are self-explanatory. It is emphasized that the Table 2-2 SQA Evaluation Plan procedures would be applied in review of documentation transmitted by the code developing organization, referencing the detailed requirements from Table 3-3. In lieu of information provided by the software developers, the SQA evaluation will use documentation previously made available by the developing organization, technical reports, subject matter expert-supplied information, engineering judgment, and other knowledge.

Appendix E describes an earlier estimate made on SQA upgrades for MACCS2 based on the SNL set of requirements. It is provided as an approximate indication of resource requirements for improvements to one of the six designated toolbox codes.

Table 4-1. SQA Evaluation of Toolbox Codes - Actions by Organization

SQA Plan Actions	Responsible Organization		
	Code Developer	SQA Evaluator	DOE/EH
1. Establishes SQA Criteria for Safety Analysis Toolbox Codes. **[SECTION 3]**		Performs	Approves
2. Provides requested documentation on SQA used in the development of subject software. Estimates resources needed to support: i) SQA deficiencies; ii) corrections to software; iii) improvements to software.	Performs	-	-
3. Apply Table 2-1 SQA Plan [SECTION 2] to Review documentation and evaluate software against SQA Criteria and implementation procedures (Table 3-3). **[SECTION 3]**	-	Performs	-
4. Document code review in "gap" analysis reports.	Reviews	Performs	-
5. Determine minimum required actions to be taken before software meets SQA Criteria	-	Recommends	Approves
6.a SQA Documentation Upgrade 6.b Software Modification – Deficiencies/Improvements	Performs; Performs	Approves	-
7. Identify Upgraded Computer Code Version for DOE Users to Central Registry	Performs	-	-
8. Provide Configuration Management and Control of Qualified Software	-	-	Note A.

Note A.
DOE does not control development of any of the designated toolbox software for safety analysis applications. However, through the Central Registry organization, it will work closely with the software developers to provide input from DOE users on software issues as well as maintain oversight on configuration management and control on versions used by the DOE user community.

5.0 Acronyms

AEC	Atomic Energy Commission
ANS	American Nuclear Society
ANSI	American National Standards Institute
ASME	American Society of Mechanical Engineers
CCPS	Center for Chemical Process Safety
CFR	Code of Federal Regulations
DNFSB	Defense Nuclear Facilities Safety Board
DoD	Department of Defense
DOE	Department of Energy
DSA	Documented Safety Analysis
EIA	Electronic Industries Alliance
EPRI	Electric Power Research Institute
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
NRC	Nuclear Regulatory Commission
OCRWM	Office of Civilian Radioactive Waste Management
QAP	Quality Assurance Program (alternatively, Plan)
SNL	Sandia National Laboratories
SQA	Software Quality Assurance
SRS	Savannah River Site
V&V	Verification and Validation
WSRC	Westinghouse Savannah River Company
YMP	Yucca Mountain Project

6.0 Definitions

The following definitions are taken from 10 CFR 830, the Implementation Plan, and other sources. References in brackets following definitions indicate the original source, when not the Implementation Plan.

Central Registry – An organization designated to be responsible for certification of the quality assurance programs of the Department's safety analysis "toolbox codes". The central registry will assist in the long-term maintenance and control of the DOE safety analysis toolbox codes. It may also perform this function for other codes if the Department determines that this is appropriate.

Documented Safety Analysis – A documented analysis of the extent to which a nuclear facility can be operated safely with respect to workers, the public, and the environment, including a description of the conditions, safe boundaries, and hazard controls that provide the basis for ensuring safety. [10 CFR 830]

Existing Software – Software that has been developed using an engineering methodology that is not fully compliant with ASME NQA-1-2000 requirements, or other related quality assurance programs.

Firmware - The combination of a hardware device and computer instructions and data that reside as read-only software on that device. [IEEE Standard 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology*]

Gap Analysis – Evaluation of the Software Quality Assurance attributes of a computer code against established requirements.

Graded Application or Approach – The process by which the level of detail in analyses, documentation, and actions necessary to comply with requirements is commensurate with-

- the relative importance to safety, safeguards, and security;
- the magnitude of any hazard involved;
- the life-cycle stage of a facility
- the programmatic mission of a facility
- the particular characteristics of a facility;
- and any other relevant factors [DOE O 414.1A].

I&C Software – Software used for instrumentation and controls (I&C) including embedded microprocessors, distributed control systems, supervisory control and data acquisition systems (SCADA), programmable logic controller (PLC), and other related software.

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 by 10 CFR 830 [10 CFR 830].

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

Quality Assurance – All those actions that provide confidence that quality is achieved. [10 CFR 830]

Quality Assurance Program – The overall program or management system established to assign responsibilities and authorities, define policies and requirements, and provide for the performance and assessment of work. [DOE O 414.1A]

Safety Analysis Software Group (SASG) – A group of technical experts formed by the Deputy Secretary in October 2000 in response to Defense Nuclear Facilities Safety Board (DNFSB)Technical Report 25. The SASG was responsible for determining the safety analysis and instrument and control (I&C) software that is widely used in the DOE Complex and required upgrade, establishing plans and cost estimates for remedial work, providing recommendations for permanent storage of the software, and coordinating with the Nuclear Regulatory Commission on code assessment as appropriate.

Safety Software – Includes both safety system software, and safety analysis and design software, as defined in the Implementation Plan for addressing DNFSB Recommendation 2002-1.

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). This also includes computer software such as human-machine interface software, network interface software, programmable logic controller (PLC) programming language software, and safety management databases that are not part of an SSC but whose operation or malfunction can directly affect SS and SC SSC functions [DOE N 411.1]

Software Evaluation Package (SEP) – A set of documents that is utilized to demonstrate adequate confidence that the existing or acquired software is acceptable for its intended end use.

Software - Computer programs, operating systems, procedures, and possibly associated documentation and data pertaining to the operation of a computer system. [IEEE Standard 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology*]

Toolbox Codes – A small number of standard computer models (codes) supporting DOE safety analysis, having widespread use, and of appropriate qualification that are maintained, managed, and distributed by a central source. These codes are verified and validated and are applicable to supporting 10 CFR 830 DSAs. The analysts using these codes do not need to present additional defense as to their qualification, provided that they are sufficiently qualified to use the codes and the input parameters are valid. It may also include commercial or proprietary design codes where DOE considers additional SQA controls are appropriate for

repetitive use in safety applications and there is a benefit to maintain centralized control of the codes [modified from DOE N 411.1].

7.0 References

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

ASME American Society of Mechanical Engineers, (2002). *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.

Bixler, N. (2000). Proposal to Resolve QA Deficiencies in MACCS2, Sandia National Laboratories, Albuquerque, NM.

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

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).

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

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

DOE, U.S. Department of Energy (2000a). *Appendix A, Evaluation Guideline,* DOE-STD-3009-94, *Preparation Guide for U.S. Department of Energy Nonreactor Nuclear Facility Safety Reports* (January 2000).

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).

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

DOE, U.S. Department of Energy (2002b). *Selection of Computer Codes for DOE Safety Analysis Applications*, National Nuclear Security Administration (August 2002).

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).

DOE/EH, U.S. Department of Energy Office of Environment, Safety and Health (2003), *Designation of Initial Safety Analysis Toolbox Codes*, Letter, (March 28, 2003).

OCRWM, Office of Civilian Radioactive Waste Management (2003a). Procedure AP-SI.1Q, Software Management, Revision 5 ICN 2, (September 2003).

OCRWM (2003b). Procedure AP-SI.2Q, Qualification of Level A Software, Revision 1 ICN 1, (September 2003).

OCRWM (2003c). Procedure AP-SI.3Q, Software Independent Verification and Validation, (September 2003).

SNL, Sandia National Laboratories (2003). Nuclear Waste Management Program Procedure, NP 19-1 Software Requirements, Revision 10, (May 29, 2003).

WSRC, Westinghouse Savannah River Company (1998). WSRC-RP-98-00712.

WSRC Westinghouse Savannah River Company (2003) 1Q Manual, QAP 20-1, Software Quality Assurance (2003).

WSRC Westinghouse Savannah River Company (2003a). E7 Manual, Conduct of Engineering and Technical Support, Section 5.0, Software Engineering and Control (July 2003).

WSRC Westinghouse Savannah River Company (2003b). E7 Manual, Software Classification (U), Procedure 5.05 (July 2003).

WSRC Westinghouse Savannah River Company (2003c). E7 Manual, Evaluation of Existing and Acquired Software (U), Procedure 5.07 (July 2003).

WSRC Westinghouse Savannah River Company (2003d). E7 Manual, Conduct of Engineering and Technical Support, Glossary (U), Revision 34 (September 2003).

8.0 Bibliography of Standards Referenced in Nuclear Quality Assurance and Software Quality Assurance

American National Standards Institute and American Nuclear Society

American National Standard, *Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry*, ANSI/ANS-10.4-1987.

American Society of Mechanical Engineers (ASME)

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.

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

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.

Center for Chemical Process Safety

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).

Code of Federal Regulations

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

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

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

10 CFR 830, *Nuclear Safety Management.*

Department of Defense (DoD)

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

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

Related guidance document, "Verification, Validation and Accreditation (VV&A) Recommended Practices Guide".

Department of Energy

DOE O 200.1 Information Management

DOE O 414.1A, Quality Assurance

DOE G 200.1-1, Software Engineering Methodology

DOE G 414.1-2, Quality Assurance Management System Guide

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

Electric Power Research Institute

1.1. EPRI TR-102348, Guideline on Licensing Digital Upgrades

1.2. EPRI NP-5652-Guidelines for the Utilization of Commercial Grade Items in Nuclear Safety-Related Applications

Institute of Electrical and Electronics Engineers (IEEE)

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

IEEE Standard 729, *IEEE Standard Glossary of Software Engineering Terminology*.

IEEE Standard 730, *IEEE Standard for Software Quality Assurance Plans*

IEEE Standard 730.1, *IEEE Guide for Software Quality Assurance Planning*

IEEE Standard 828, *IEEE Standard for Software Configuration Management Plans*

IEEE Standard. 829, *IEEE Standard for Software Test Documentation*.

IEEE Standard 830, *Software Requirements Specifications*

IEEE Standard 1008, *Software Unit Testing*

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 1063, *IEEE Standard for Software User Documentation*

IEEE Standard 1074, *IEEE Standard for Developing Software Life Cycle Processes*

IEEE/EIA Standard 12207.0, *Industry Implementation of International Standard ISO/IEC 12207 Standard for Information Technology – Software Life Cycle Processes*

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

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

International Atomic Energy Agency (IAEA)

IAEA Safety Guide 50-SG-D3, *Protection Systems and Related and Related Systems*.

IAEA Safety Guide 50-SG-D8, *Safety-Related Instrumentation and Control Systems*.

IAEA Safety Guide 50-SG-Q1, *Establishing and Implementing a Quality Assurance Programme*.

International Electrotechnical Commission (IEC)

IEC 880, Software for Computers in the Safety Systems of Nuclear Power Stations

IEC 987, Programmed Digital Computers Important to Safety for Nuclear Power Stations

IEC 1226, Nuclear Power Plants – Instrumentation and Control Systems Important for Safety – Classification.

IEC 9126, Information Technology - Software Product Evaluation – Quality Characteristics and Guidelines for Their Use

IEC 12207, Information Technology – Software Life-Cycle Processes

International Organization for Standardization (ISO)

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*

Appendix A. Software Classification Levels and Graded Application

The primary SQA criteria (NQA-1a-1999 and NQA-1-2000) discussed in Section 3 recommend that the level of SQA associated with a computer code be commensurate with the importance of the software application. This is characteristic of a graded application or approach. Each of the three SQA programs is based on functional classification basis to establish similar requirements for software. Table A-1 cross-references the software classification level with its historical, functional classification basis.

Table A-1. – Software Functional Classification Level Cross-Referenced to SSC Functional Classification[7]

Software Classification Level	**Functional Classification Basis**	**Definition**
A	Safety Class (SC)	The functional and safety classification that applies to those structures, systems, or components or administrative controls whose preventative or mitigative function is necessary to for protection of the public.
B	Safety Significant (SS)	The functional and safety classification that applies to those structures, systems, or components or administrative controls not designated as Safety Class SSCs but whose preventive or mitigative function is a major contributor to defense in depth and/or worker safety.
C	Production Support (PS)	The functional classification that applies to those SSCs necessary to support continued operation of a nuclear facility and to selected environmental monitoring and emergency plan communications devices.
D	General Services (GS)	The functional classification assigned to those SSCs not required to provide a Safety Class, Safety Signification, or Production Support function. As a minimum, SSCs at this level are to assure proper design, operations, and maintenance requirements are met to provide for the health and safety of the worker and environment, and to assure compliance with local (laboratory/site) requirements.
E	General Services (GS)	Same as immediately above.

[7] Based in part on WSRC (2003d).

Appendix B. Criteria for Software Quality Assurance for Safety-Related Software Applied to DOE Nuclear Facilities

Prior to initiating an evaluation of the compliance of the candidate toolbox software with current software standards, one or more reference benchmark sets of criteria must be identified. This Appendix will summarize quality assurance criteria and requirements, SQA standards, and related criteria from the nuclear and software quality sectors.

B.1 10 CFR 830, Subpart A – Quality Assurance Requirements

Subpart A to 10 CFR 830 establishes quality assurance requirements for DOE contractors conducting activities including providing items or services, that affect, or may affect, the nuclear safety of DOE nuclear facilities. Section 830.121 describes a requisite quality assurance program (QAP), its applicability, frequency of updates, and directs the contractor to describe how criteria (Section 830.122) are met. It also specifies integration with the Safety Management System and recommends use of voluntary consensus standards.

Ten broad quality assurance criteria are described in Section 830.122. Each quality assurance criterion is stated as a performance expectation without specification of the methods for achieving the desired result. Instead, contractors are directed to national and international standards to develop effective and efficient QAPs. The management, performance, and assessment criteria include:

1 – Management Program
2 - Management/Personnel Training and Qualification
3 - Management/Quality Improvement
4 – Management/Documents and Records
5 – Performance/Work Processes
6 – Performance/Design
7 – Performance/Procurement
8 – Performance/Inspection and Acceptance Testing
9 – Assessment/Management Assessment
10 – Assessment/Independent Assessment.

The DOE implementation guide for quality assurance requirements from the 10 CFR 830 rule is DOE G 414.1-2. The latter includes a discussion of standards use, and references the most widely accepted standards for quality assurance.

B.2 Department of Energy Directives

B.2.1 Department of Energy Order 414.1A, *Quality Assurance* (cancels DOE O 414.1)

DOE O 414.1A describes how to establish an effective management system, in terms of quality assurance programs, using the performance requirements identified in 10 CFR 830.122 and reiterated in the Order. Coupled with technical consensus standards where appropriate, DOE O 414.1A will ensure:

- senior management provides planning, organization, direction, control, and support to achieve DOE objectives,
- line organization functions and responsibilities are defined, and
- each DOE element reviews, evaluates and improves its overall performance using a rigorous assessment process.

This order also states that DOE elements must implement the quality assurance criteria in a manner sufficient to achieve adequate protection of workers, public and the environment, taking into account the work to be performed and the associated hazards. Quality assurance criteria in 10 CFR 830.122 must be used to develop the QAPs, with the latter describing how the criteria are satisfied. Use of the graded approach must be described.

B.2.2 Department of Energy Notice 411.1, *Safety Software Quality Assurance Functions, Responsibilities, and Authorities for Nuclear Facilities and Activities*

DOE N 414.1A assigns roles and responsibilities for improving the quality of safety software. Responsibilities are assigned to the: (a) Assistant Secretary for Environment, Safety, and Health; (b) Chief Information Officer; (c) Secretarial Officers; (d) Field Element Managers; and (e) Office of Independent Oversight.

B.2.3 Department of Energy Guide 414.1-1, *Implementation Guide for Use with Independent and Management Assessment Requirements of 10 CFR 830.120 and DOE 5700.6C Quality Assurance*

Subpart A to 10 CFR 830 and the DOE QA order establish requirements for DOE and its contractors to perform Management and Independent Assessments using appropriate standards wherever applicable. DOE G 414.1 provides information concerning the establishment and implementation of effective assessment processes.

B.2.4 Department of Energy Guide 414.1-2, *Quality Assurance Management System Guide for use with 10 CFR 830.120 and DOE O 414.1*

DOE G 414.1-2 provides information on principles, requirements, and practices used to establish and implement an effective Quality Assurance Program (QAP or quality management system) in accordance with the requirements of 10 CFR Subpart A (previously 10 CFR 830.120) and DOE O 414.1 (superseded by DOE O 414.1A). The Guide also describes the relationship of quality assurance to other processes that aid compliance with Integrated Safety Management System (ISMS) requirements.

The Guide assists the user in obtaining DOE customer concurrence on the QAP. It is stated in the introduction to the guide that implementation will contribute to improved safety, management, and the reliability of DOE products and services. DOE G 450.4-1B, The Integrated Safety Management System Guide, contains more information on safety management principles, supporting attribute, and references on the subject. Guidance is provided on the same ten criteria enumerated in 10 CFR 830 Subpart A.

B.2.5 Department of Energy Policy 450.4, *Safety Management System Policy*

DOE P 450.4 is a top-level document that describes Safety Management Systems as providing a formal, organized process whereby people plan, perform, assess, and improve the safe conduct of work. The Safety Management System is institutionalized through Department of Energy (DOE) directives and contracts to establish the Department-wide safety management objective, guiding principles, and functions. The Policy outlines the basic tenets of the safety management system through Integrated Safety Management: 1- Objective; 2 – Guiding Principles; 3 – Core Functions; 4 – Mechanisms; 5 – Responsibilities; and 6 – Implementation.

B.2.6 Department of Energy Order 200.1, *Information Management Program*

DOE O 200.1 is a corporate order that was developed to improve the quality and usability of the policies and requirements associated with DOE's various information management functions. The order indicates that information management activities shall be established, maintained, and managed in a manner that addresses Departmental policy and implements appropriate laws and regulations. The latter are itemized as an attachment to the Order.

B.2.7 Others

The following are noted in the remainder of this document but were not extensively used:

- Department of Energy Office of Civilian Radioactive Waste Management, Quality Assurance Requirements and Description, DOE/RW-0333P, Revision 11
- Quality Control for Weapons Programs, QC-1
- EPRI NP-5652-Guidelines for the Utilization of Commercial Grade Items in Nuclear Safety-Related Applications
- EPRI TR-102348, Guideline on Licensing Digital Upgrades.

B.3.1 Quality Assurance in the Commercial Nuclear Sector[8]

10 CFR Parts 50 and 70

It is noted that early use of quality assurance in the naval reactor program and then in commercial nuclear industry evolved from quality control practices. By the late 1960s, the Atomic Energy Commission (AEC) integrated naval reactor program practices with those of the National Aeronautics and Space Administration (NASA) and the Department of Defense (DoD) in issuing early guidance. In 1970, the AEC issued 10 CFR 50, Appendix B, *Quality Assurance for Nuclear Power Plants*. Since the inception of the Nuclear Regulatory Commission (NRC), Appendix B has been modified but the basic set of elements required for quality assurance programs has been consistent. Several categories of facilities are discussed in terms of U.S. Nuclear Regulatory Commission (NRC) SQA requirements: power and fuel reprocessing plants, special nuclear material, and the Yucca Mountain project.

Under the requirements of Appendix B to 10 CFR Part 50, Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants, every commercial plant and fuel processing plant applicant (for a construction permit) is required by the provisions of §50.34 to include in its preliminary safety analysis report a description of the quality assurance program to be applied to the design, fabrication, construction, and testing of the structures, systems, and components of the facility. Every applicant for an operating license is required to include, in its final safety analysis report, information pertaining to the managerial and administrative controls to be used to assure safe operation. Nuclear power plants and fuel reprocessing plants include structures, systems, and components that prevent or mitigate the consequences of postulated accidents that could cause undue risk to the health and safety of the public. Quality assurance requirements for the design, construction, and operation of those structures, systems, and components are established in Appendix B to 10 CFR 50. The requirements of Appendix B apply to all activities affecting the safety-related functions of those structures, systems, and components (including designing, purchasing, fabricating, handling, shipping, storing, cleaning, erecting, installing, inspecting, testing, operating, maintaining, repairing, refueling, and modifying).

As used in Appendix B, "quality assurance" comprises all those planned and systematic actions necessary to provide adequate confidence that a structure, system, or component will perform satisfactorily in service. Quality assurance in this case includes quality control, i.e. those actions related to the physical characteristics of a material, structure, component, or a system that provide a means to control the quality of the material, structure, component, or system to predetermined requirements.

Although these sections do not specifically address software quality, it does indicate that QA criteria apply to activities (e.g., design, test, operation, and modification) associated with safety-related functions of structures, systems, and components important to safety. By inference, software used to support such activities would be expected to be subject to the applicable requirements of Subpart B.

---

[8] Much of this discussion is based on DNFSB/TECH-31, Engineering Quality into Safety Systems (DNFSB, 2001).

Similar arguments for judicious application of SQA to software-supported safety-related design and analysis activities may be interpreted from Part 70 (Domestic Licensing of Special Nuclear Material) and Section 142 (Quality Assurance Criteria) to Part 63 (Disposal of High-Level Radioactive Wastes in a Geologic Repository at Yucca Mountain, Nevada). In the latter example, the DOE QA program for the Yucca Mountain site characterization includes a discussion of how the applicable requirements of Appendix B are met (DOE 1998). The QA program contains controls for software that appear in ASME NQA-1-1997, Part II, Section 2.7, such as: (1) software life cycles, baselines, and controls; (2) software verification and validation; (3) software configuration management; (4) defect reporting and resolution; (5) control of the use of the software; and (6) software documentation. The NRC has reviewed this plan and found it acceptable.

NQA-1 and Successor Criteria

In the mid-1970s, the American National Standards Institute (ANSI) assigned overall responsibility for coordination among technical societies and the development and maintenance of standards for Nuclear Quality Assurance (NQA) to the American Society of Mechanical Engineers (ASME). Subsequently, the ASME Committee on Quality Assurance prepared ANSI/ASME NQA-1 (*Quality Assurance Program Requirements for Nuclear Power Plants*), and ANSI/ASME NQA-2 (*Quality Assurance Requirements for Nuclear Power Plants)* in 1979 and 1983, respectively.

The NQA-1 standard was revised and reissued several times in the period of 1983 – 1997. NQA-2 was written for a number of systems not explicitly treated in NQA-1. It was revised and reissued in 1986 and 1989. In the 1980s, an NQA subcommittee on Nuclear Waste Management prepared ASME NQA-3, Quality Assurance Program Requirements for the Collection of Scientific and Technical Information for Site Characterization of High-Level Nuclear Waste Repositories.

In the 1990s, the NQA Committee of ASME decided to merge the three NQA standards into one, multi-part document. The 1997 version of NQA-1 was structured as follows:

- I – former NQA-1, including quality assurance program requirements for the siting, design, construction, operation, and decommissioning of nuclear facilities
- II – former NQA-2, including the quality assurance requirements for the planning and execution of identified tasks during the fabrication, construction, modification, repair, maintenance, and testing of systems, components, and structures for nuclear facilities
- III – non-mandatory guidance and application appendices
- IV – future NQA position papers and application matrices.

Two revisions have been approved recently. ASME NQA-1a-1999 contains a complete revision to Subpart 2.7, Quality Assurance Requirements for Computer Software for Nuclear Facility Applications. While format is modified and consolidated from the earlier version, pertinent references from the Institute of Electrical Engineers were also used in the update. ASME NQA-1 2000 is the most recent revision. However, Subpart 2.7 is unchanged from the 1999 version.

B.3.2 International Quality Management Standard

The International Organization for Standardization (ISO) is a worldwide federation of national standard bodies. ISO has prepared a series of standards emphasizing quality management practices. ISO 9001*, Quality systems -- Model for quality assurance in design, development, production, installation and servicing*, was issued in 1994. It specifies quality system requirements to be used when a supplier's capability to design and supply parts needs to be demonstrated.

ISO 9001 did not specifically address computer software. *ISO Quality management and quality assurance standards - Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software*, ISO 9000-3, provides quality guidelines to help organizations to apply the ISO 9001:1994 requirements to computer software. ISO 9000-3 is intended for computer software developers, suppliers, installers, and maintainers. ISO 9000-3:1997 is an expanded version of the ISO 9001 standard. The new standard adapts ISO 9001 by adding some new text that refers only to software.

The most recent ISO standard in this series, ISO 9001-2000, *Quality management systems – requirements*, promotes adoption of a process approach when developing, implementing and improving the effectiveness of a quality management system, to enhance customer satisfaction by meeting customer requirements. It is noted that the computer software-specific ISO 9000-3 standard has not since been revised to correspond to the new ISO 9001.

B.3.3 Defense Software Development and Maintenance

The applicable standard for many Department of Defense (DoD) software projects is Directive 5000.61 and related guidance. Other national and industry standards are implemented to guide and support software work as appropriate.

The overall framework and general format for the independent review to be applied to the Subject Software is outlined in the DoD Instruction 5000.61 guidance document *Verification, Validation and Accreditation (VV&A) Recommended Practices Guide*. The Guide uses the terminology of running a computer model as a simulation. Hence executing a software package is termed a Modeling and Simulation (M&S) of a process or resulting phenomenology.

Independence is encouraged to preserve distinct separation and responsibilities among software designer, user, and review groups for formal verification, validation, and accreditation (VV&A). The intent of independent VV&A is to provide the <u>documented</u> assurance that selected M&S not only provides output that meets the expectations of the design, but that the M&S being used is applicable to the specific function required of the software.

The VV&A steps depend on the accreditation status of the subject software and previous use. Specifically, the particular VV&A process depends on which of the following categories the applicable M&S falls into:

1) previously accredited based on verification and validation data, which is available;
(2) previously accredited based on historical use;
(3) not previously accredited, but some verification and validation data available; and
(4) not previously accredited, with little or no verification and validation available.

More recent software development solicitations from applied research and development organizations in DoD have specified industry standards. For example, the Defense Threat Reduction Agency (DTRA) has cited ISO/IEC Standard 12207 (***Standard for Information Technology - Software Life Cycle Processes***) in software development requests for proposal. ISO/IEC 12207 was initially published in 1995 and provided a comprehensive set of life cycle processes, activities and tasks for software that is part of a larger system, stand alone software product, and software services. The standard provides common software process framework for the acquisition, supply, development, operation and maintenance of software. The standard also provides the necessary supporting processes, activities and tasks, and organizational processes, activities and tasks for managing and improving the processes.

Amendment 1 is an interim revision to ISO/IEC. The Amendment accommodates the requirements of current and developing standards and technical reports, including ISO/IEC 12207 and ISO/IEC/TR 15504, as well as considering other standards, e.g., ISO/IEC 14598 and ISO/IEC 15939.

Industry implementation of ISO/IEC Standard 12207 is through:

- IEEE/EIA 12207.0-1996 (**Standard for Information Technology - Software Life Cycle Processes)**
- IEEE/EIA 12207.1 (**Industry Implementation of ISO/IEC 12207:1995 - Standard for Information Technology - Software Life Cycle Processes - Life Cycle Data), and**
- IEEE/EIA 12207.2-1997 (**Industry Implementation of International Standard ISO/IEC 12207: 1995; Standard for Information Technology - Software Life Cycle Processes - Implementation Considerations**).

B.3.4 International Commercial Nuclear and Software Standards

Alternative models can be summarized from the International Atomic Energy Agency (IAEA), the International Electrotechnical Commission (IEC), and the International Organization for Standardization (ISO). For example, in the nuclear power plant standards area, 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) are linked to IAEA Safety Guides, which in turn, address IAEA Safety Standards (Scott, 1996). Similarly, IEC works closely with ISO in developing substantive engineering standards for software development and system documentation. The ISO-IEC relationship is analogous to the one between American National Standard Institute (and ASME), and the Institute of Electrical and Electronic Engineers in the U.S.

B.3.5 National Aeronautics and Space Administration

The National Aeronautics and Space Administration (NASA) defines SQA as a planned and systematic approach to the evaluation of the quality of and adherence to software product standards, processes, and procedures. SQA includes the process of assuring that standards and procedures are established and are followed throughout the software acquisition life cycle. Compliance with agreed-upon standards and procedures is evaluated through process monitoring, product evaluation, and audits. Software development and control processes shall include quality assurance approval points, where an SQA evaluation of the product shall be done in relation to applicable standards. The key software standard required for compliance is the NASA Software Assurance Standard, NASA-STD-2201-93.

B.3.6 Comparisons of Standards

Comparisons among sets of standards are available and are typically updated after new revisions have been issued. ASME NQA Technical Report, Comparison of ASME NQA-1 and ISO 9001, (September 1993) is a useful comparison comparing the two standards on the 18 primary requirements covered in NQA-1, and noting the ISO 9001 Requirements without NQA-1 counterparts (contract review, servicing, and statistical techniques). Currently, the ASME NQA Committee is in the process of reviewing the requirements of ISO 9001-2000 to determine what additional controls or requirements would be needed in order to meet ASME NQA-1-2000 appropriate for nuclear facilities/activities and their oversight (ASME, 2002).

Appendix B to DNFSB/TECH-31 contains a detailed comparison of the major provisions addressed in 10 CFR 50, Appendix B, NQA-1-1994, 10 CFR 830.120 (superseded by 10 CFR 830, Subpart A), and ISO 9001 (DNFSB 2001).

Appendix C.

Table C-1. Comparison of Subpart 2.7 from NQA-1-1997 with NQA-1a-1999 and NQA-1-2000

Section (1997 version)	NQA-1-1997; Subpart 2.7	NQA-1-2000; Subpart 2.7[9]
100	General Indicates that Subpart 2.7 provides requirements for the development, procurement, maintenance, and use of computer software, as applied to the design, construction, operation, modification, repair, and maintenance of nuclear facilities. Supplements requirements of Part I and shall be used in conjunction with applicable sections of Part I when and to the extent specified by the organization invoking Subpart 2.7.	100/General Indicates that Subpart 2.7 provides requirements for the acquisition, development, operation, maintenance, and retirement of software. Appropriate requirements shall be implemented through the policies, procedures, plans, specifications, or work practices, etc. that provide the framework for software engineering activities. Supplements requirements of Part I and shall be used in conjunction with applicable sections of Part I when and to the extent specified by the organization invoking Subpart 2.7.
101	Definitions The following are defined as used in Subpart 2.7: baseline, code, computer program, configuration control, configuration item, error, portability, software, software life cycle, software quality assurance plan, software validation, software verification, systems software, test case, test plan, testing. -	102/Definitions The following are defined as used in Subpart 2.7: acceptance testing, baseline, configuration management/software, configuration item, control point, error, operating environment, software design verification, software development cycle, software engineering, software life cycle, software tool, system software, testing (software) – includes: a) operating a system or system component under specified conditions; b) observing and recording the results; and c) making an evaluation of some aspect of the system (i.e. software and hardware) or system component; in order to verify that it satisfies specified requirements and to identify errors. Also, test case and test plan/procedure are defined.
102		101/Software Engineering Scope of software engineering activities include the following elements, as appropriate: a) software acquisition method(s) for controlling the acquisition process for software and software services; b) software engineering method(s) used to manage the software life-cycle activities; c) application of standards, conventions, and other work practices that support the software life cycle; and d) controls for support software used to develop, operate, and maintain computer programs.
200	General Requirements	200/General Requirements The following general requirements shall be applied to the software engineering elements described in Paragraph 101 of Subpart 2.7.
201	Applicability Specifies that the requirements in 2.7 apply to computer software used to produce or manipulate data, used directly in the design, analysis, and operation of structures, systems, and components. Application of specific requirements shall be prescribed in plans for software quality assurance and in written policies and procedures.	-

[9] Subpart 2.7 did not change from NQA-1A-1999 to NQA-1-2000. In terms of intent, it still supplements requirements of Part 1 and is used in conjunction with applicable requirements of Part 1 when and to the extent specified by the organization invoking the Subpart. Requirements from Part 1 are listed in parentheses in the section of 2.7 where the reference is made.

Table C-1. Comparison of Subpart 2.7 from NQA-1-1997 with NQA-1a-1999 and NQA-1-2000 (Continued)

Section (1997 version)	NQA-1-1997; Subpart 2.7	NQA-1-2000; SUBPART 2.7[10]
300	Software Life Cycle Subpart 2.7 based on life cycle model similar to IEEE Standard 1012 – systematic approach to software development and maintenance. Intent is not to endorse or restrict any particular model, provided it encompasses activities associated with the representative software life cycle shown in Figure 300 –Requirements, Design, Implementation, Test, Installation and Checkout, Operation and Maintenance, Retirement. Notes software development shall proceed in traceable, planned and orderly manner. The number of phases and relative emphasis placed on each phase of software development will depend on nature and complexity of software. Software development may be performed in an iterative or sequential manner.	400/Software Engineering Method Software engineering method(s) shall be documented. Selected software engineering method shall ensure that software life cycle activities are planned and performed in a traceable and orderly manner. The requirements of Part I, Requirement 3 shall be met. (Part 1, Requirement 3 – See Table C-3)
301	Requirements Phase Requirements that software must satisfy that pertain to functionality, performance, design constraints, attributes, and external interfaces (outlined in Paragraph 602) shall be specified, documented, and reviewed. The requirements shall define the response of the software to anticipated classes of input data, and shall provide the detail and information necessary to design the software. Requirements phase activities include the preparation of plans for software verification and validation typically called the software verification and validation plan.	401/Software Design Requirements Software design requirements shall address technical and software engineering – paragraph 101 of Subpart 2.7 – requirements. Software design requirement shall be traceable throughout the software life cycle.
302	Design Phase Software design based on the requirements shall be developed , documented , and reviewed. The design shall specify the overall structure 9control and data flow) and the reduction of the overall structure into physical solutions (algorithms, equations, control logic, and data structures). The design may necessitate the modification of the requirements documentation. Design phase software verification and validation shall consist of: a) generation of test plans based on the requirements and design; b)generation of design-based test cases; c) review of the software design to ensure that the requirements are addressed.	402/Software Design Integral part of software design is the design of a computer program that is part of an overall system. The software design shall consider the computer program's operating environment. Measures to mitigate the consequences of problems shall be an integral part of the design. These potential problems include external and internal abnormal conditions and events that can affect the computer program. 402.1 Software Design Verification. Software design verification shall evaluate the technical adequacy of the design approach and assure internal completeness, consistency, clarity, and correctness of the software design and shall verify that software design is traceable to the software design requirements. Software design verification shall include review of test results. Shall be completed prior to approval of the computer program for use. Requirements for the software design verification activity shall be documented in the software engineering method.

[10] Subpart 2.7 did not change from NQA-1A-1999 to NQA-1-2000. In terms of intent, it still supplements requirements of Part 1 and is used in conjunction with applicable requirements of Part 1 when and to the extent specified by the organization invoking the Subpart. Requirements from Part 1 are listed in parentheses in the section of 2.7 where the reference is made.

Table C-1. Comparison of Subpart 2.7 from NQA-1-1997 with NQA-1a-1999 and NQA-1-2000 (Continued)

Section (1997 version)	NQA-1-1997; Subpart 2.7	NQA-1-2000; Subpart 2.7[11]
303	Implementation Phase The designs shall be translated into a programming language, and the implemented software shall be analyzed to identify and correct errors. Implementation phase software verification activities shall consist of the examination of computer program listings to assure adherence to coding standards and conventions.	403/Implementation This process shall result in software products such as computer program listings and instructions for computer program use. A review shall be performed in accordance with Paragraph 202 of Subpart 2.7.
304	Testing Phase The design as implemented in code shall be exercised by executing the test cases. Failure to successfully execute test cases shall be reviewed to determine if modifications of the requirements, the design, the implementation, or the test plans and test cases are required. Testing phase activities shall consist of the validation of the code to assure adherence to the requirements, and to assure that the software produces correct results for the test cases. To evaluate technical adequacy, the software test case results can be compared to results from alternative methods, such as: a) analysis without computer assistance; b)other validated computer program; c) experiments and tests; d) standard problems with known solutions; or e) confirmed published data and correlations. See Part I, requirement 11, Test Control.	404/Acceptance Testing (Part 1, Requirement 11; See Table C-4) Acceptance testing activity shall demonstrate that the computer program adequately and correctly performs all intended functions – specified software design requirements. Acceptance testing shall demonstrate, as appropriate, that the computer program: a) properly handles abnormal conditions and events as well as credible failures; b) does not perform adverse unintended functions; and c) does not degrade the system either by itself, or in combination with other functions or configuration items. Acceptance testing shall be performed prior to approval of the computer program for use. Configuration items shall be under configuration change control prior to starting acceptance testing. Acceptance testing shall be planned and performed for all software design requirements. Acceptance testing shall be planned and performed for all software design requirements. Acceptance testing ranges from a single test of all software design requirements to a series of tests performed during computer program development. Performance of a series of tests provides assurance of correct translation between activities and proper function of individual modules. Testing shall include a comprehensive acceptance test performed in the operating environment prior to use. Test plans, test cases and test results shall be documented, reviewed and approved prior to use of the computer program in accordance with Part I, Requirement 11. Observations of unexpected or unintended results shall be documented and dispositioned prior to test result approval. Acceptance testing of changes to the computer program shall be subjected to selective retesting to detect unintended adverse effects introduced during the change. Such testing shall provide assurance that the changes have not caused unintended adverse effects in the computer program, and to verify that a modified system(s) or system component(s) still meets specified software design requirements.

[11] Subpart 2.7 did not change from NQA-1A-1999 to NQA-1-2000. In terms of intent, it still supplements requirements of Part 1 and is used in conjunction with applicable requirements of Part 1 when and to the extent specified by the organization invoking the Subpart. Requirements from Part 1 are listed in parentheses in the section of 2.7 where the reference is made.

Table C-1. Comparison of Subpart 2.7 from NQA-1-1997 with NQA-1a-1999 and NQA-1-2000 (Continued)

Section (1997 version)	NQA-1-1997; Subpart 2.7	NQA-1-2000; SUBPART 2.7[12]
305	Installation and Checkout During this phase, software becomes part of a system incorporating applicable software components, hardware, and data. The process of integrating the software with applicable components may consist of installing hardware, installing the program, reformatting or creating databases, and verifying that all components have been included. Installation and checkout phase software verification and validation activities shall consist of; a) execution of tests for installation and integration; and b) documentation of the approval of the software for operational use.	-
306	Operation and Maintenance Phase Before this phase the software has been approved for operational use. Further activity shall consist of maintenance of the software to remove latent errors (correct maintenance), to respond t new or revised requirements (perfective maintenance), or to adapt the software to changes in the operating environment (adaptive maintenance). Software modifications shall be approved, documented, verified and validated, and controlled. In-use tests shall be performed in accordance with Requirement 11 of Part I.	405/Operation After the software is approved for use and installed in the operating environment, the use of the software shall be controlled in accordance with approved procedures and instruction. These include, as appropriate: a) application documentation; b) access control specifications; c) problem reporting and corrective action; d) in-use tests; and e) the configuration change control process. 406/Maintenance The appropriate software engineering elements as described in paragraph 101 in Subpart 2.7 shall identify how changes to the software are controlled. Typically, changes are in response to: a) enhancement requests from the user community; b) revisions to software based on software design requirements; c) changes to the operating environment; and d) reported software problems that must be corrected.
307	Retirement Phase In the retirement phase the support for a software product is terminated, and the routine use of the software shall be prevented.	407/Retirement During retirement, support for software product is terminated, and the routine use of the software shall be prevented.
400	Software Verification and Validation Software verification and validation activities shall: a) ensure that the software adequately and correctly performs all intended functions; and b) ensure that the software does not perform any unintended function that either by itself or in combination with other functions can degrade the entire system. Software verification and validation activities shall be planned and performed for each system configuration that may impact the software. Results of the software verification and validation activities shall be documented. Software verification and validation shall be performed by individuals other than those who designed the software (verification and validation is equivalent to Requirement 3 of Part I).	

[12] Subpart 2.7 did not change from NQA-1A-1999 to NQA-1-2000. In terms of intent, it still supplements requirements of Part 1 and is used in conjunction with applicable requirements of Part 1 when and to the extent specified by the organization invoking the Subpart. Requirements from Part 1 are listed in parentheses in the section of 2.7 where the reference is made.

Table C-1. Comparison of Subpart 2.7 from NQA-1-1997 with NQA-1a-1999 and NQA-1-2000 (Continued)

Section (1997 version)	NQA-1-1997; Subpart 2.7	NQA-1-2000; SUBPART 2.7[13]
401	Software Verification Software verification shall be performed during the software development to ensure that the products of a given cycle phase fulfill the requirements of the previous phase or phases.	
402	Software Validation Software validation is performed at the end of the implementation phase to ensure that the code satisfies the requirements. Software validation activities, such as the development of test plans and test cases, shall be integrated into each phase of the software life cycle. Testing shall be the primary method of software validation. The validation of modifications shall be subject to selective regression testing to detect errors introduced during the modification of systems or system components, to verify that the modifications have not caused unintended adverse effects, or to verify that a modified system(s) or system component(s) still meets specified requirements.	
500	Software Configuration Control	203/Software Configuration Management (Part 1, Requirement 3) In addition to the requirements of Part I, Requirement 3, software configuration management activities shall include: a) appropriate software engineering elements (described in paragraph 101 of Subpart 2.7) shall identify when configuration baselines are to be established. Configuration items to be controlled shall include as appropriate: 1-documentation; 2 – computer programs; 3 – support software; and b) the software configuration change control process shall include: 1 – initiation, evaluation, and disposition of a change request; 2 – control and approval of changes prior to implementation; and 3 – requirements for retesting and acceptance of test results.
501	Configuration Identification A configuration baseline shall be defined at the completion of each major phase of the software development. Approved changes created subsequent to a baseline shall be added to the baseline. A baseline shall define the most recent approved software configuration. A labeling system for configuration items shall be implemented that: a) uniquely identifies each configuration item; b) identifies changes to configuration items by revision; and c) provides the ability to uniquely identify each configuration of the revised software available for use.	
502	Configuration Change Control Changes to software shall be formally documented. This documentation shall contain a description of the change, the rationale for the change, and the identification of affected baselines. The change shall be formally evaluated and approved by the organization responsible for the original design, unless an alternate organization has been given authority to approve the changes. Only authorized changes shall be made to software baselines. Software verification activities shall be performed for the change as necessary to ensure the change is appropriately reflected in software documentation, and to ensure that document traceability is maintained. Software validation shall be performed as necessary for the change.	

[13] Subpart 2.7 did not change from NQA-1A-1999 to NQA-1-2000. In terms of intent, it still supplements requirements of Part 1 and is used in conjunction with applicable requirements of Part 1 when and to the extent specified by the organization invoking the Subpart. Requirements from Part 1 are listed in parentheses in the section of 2.7 where the reference is made.

Table C-1. Comparison of Subpart 2.7 from NQA-1-1997 with NQA-1a-1999 and NQA-1-2000 (Continued)

Section	NQA-1-1997; Subpart 2.7	NQA-1A-1999 AND NQA-1-2000; SUBPART 2.7
503	Configuration Status Accounting Information that is needed to manage a configuration shall be documented. This information shall identify the approved configuration, the status of proposed changes to the configuration, the status of approved changes, and information to support the functions of configuration identification, and configuration control.	
600	Documentation Required documentation for software is identified in Section 600.	201/Documentation Appropriate software engineering elements, described in Paragraph 101 of Subpart 2.7 shall define the baseline documents that are to be maintained as records, in accordance with Part I, Requirement 17. Although multiple documentation requirements are specified within this Subpart, they can be provided as separate or as combined documents (Part 1, Requirement 17)
601	Plan(s) for Software Quality Assurance Plans(s) for assuring software quality assurance shall be in existence for each new software project at the start of the software life cycle, or for procured software when it enters the Purchaser's organization. The plan(s) may be prepared individually for each software project, or may exist as generic document to be applied to software prepared within or procured by an organization, or may be incorporated into the overall quality assurance program. Plan for software quality assurance shall identify: a) the software products to which it applies b) the organizations responsible for performing the work and achieving software quality and their tasks and responsibilities; c) required documentation; d) standards, conventions, techniques, or methodologies which shall guide the software development, as well as methods to assure compliance to the same; e) the required software reviews; and f) the methods for error reporting and corrective action.	500/Standards, Conventions, and Other Work Practices As appropriate, the software engineering method, software acquisition method, or both shall establish the need for standards, conventions, and other required work practices to facilitate software life cycle activities. Standards, conventions, and other required work practices shall be documented.

Table C-1. Comparison of Subpart 2.7 from NQA-1-1997 with NQA-1a-1999 and NQA-1-2000 (Continued)

Section	NQA-1-1997; Subpart 2.7	NQA-1A-1999 AND NQA-1-2000; SUBPART 2.7
602	Software Requirements Documentation Software requirements documentation shall outline the requirements that the proposed software must satisfy. The requirements shall, as applicable, address the following: a) functionality – functions the software is to perform b) performance – time-related issues of software operation such as speed, recovery time, response time, etc. c) design constraints imposed on implementation phase activities – any elements that will restrict design options d) attributes – non-time-related issues of software operation such as portability, acceptance criteria, access control, maintainability, etc.; and e) external interfaces- interactions with people hardware, and other software. An item can be called a software requirement only if its achievement can be verified and validated. Software requirements shall be traceable throughout the remaining stages of the software development cycle.	
603	Software Design and Implementation Documentation Software design and implementation documentation includes a document or series of documents that shall contain: a) a description of the major components of the software design as they relate to the software requirements; b) technical description of the software with respect to the theoretical basis, mathematical model, control flow, data flow, control logic, and data structure; c) description of the allowable or prescribed ranges for inputs and outputs; d) design described in a manner that can be translated into code; and e) computer program listing(s).	
604	Software Verification and Validation Documentation Software verification and validation documentation shall describe the tasks and criteria accomplishing the verification of the software in each phase and the validation of the software at the end of the development cycle. The documentation shall also specify the hardware and software configuration pertinent to the software verification and validation. The documentation shall be organized in a manner that allows traceability to both the software requirements and the software design. This documentation shall also contain the results of the execution of the software verification and validation activities, and shall include the results of reviews and tests, and a summary of the status of the software, e.g., incomplete design performance and application requirements.	

605	User Documentation User documentation, as a minimum, shall include: a) user instructions that contain an introduction, a description of the user's interaction with the software, and a description of any required training necessary to use the software; b) input and output specifications c) input and output formats d) a description of system limitations e) description of user messages initiated as a result of improper input and how the user can respond; f) information for obtaining user and maintenance support.	
700	Verification Reviews These reviews shall identify the participants and their specific responsibilities during the review and in the preparation and distribution of the review documentation. The reviewed documents shall be updated and placed under configuration control. Documentation of review comments and their disposition shall be retained until they are incorporated into the updated software. Comments and their disposition not incorporated shall be retained in accordance with established procedures.	202/Review The appropriate software engineering elements, described in Paragraph 101 of Subpart 2.7 shall define the control points and associated reviews. Reviews of software shall assure compliance with the approved software design requirements. Although multiple review requirements are specified in Subpart 2.7, the reviews may be performed and documented separately or combined, as appropriate, to the defined software engineering method. Two reviews are required: a) one review shall consider the requirements related to the activities of preparing the computer program for acceptance testing. This review can be combined with or be part of the software design verification. b) the other review shall provide assurance of the satisfactory completion of the software development cycle including acceptance testing. This review can be combined with or be part of software design verification. Individuals familiar with the design detail and the intended use of the computer program shall be included in the review. Reviews shall identify participants and their specific review responsibilities. Documentation of review comments and disposition shall be retained until they are incorporated into updated software. Comments not incorporated and their disposition shall be retained until the software is approved for use. When review alone is not adequate to determine if requirements are met, alternate calculations shall be used or tests shall be developed and integrated into the appropriate activities of the software development cycle. Test performed in support of a review can be used to complement acceptance testing. The tests and test results shall be included in the acceptance testing documentation. These tests shall be subjected to the same criteria as the acceptance tests. These tests do not substitute for the performing the comprehensive, end of development, acceptance test.
701	Software Requirements Review Review shall be performed at the completion of the software requirements documentation. Shall assure that the requirements are complete, verifiable, consistent, and technically feasible. The review shall also assure that the requirements will result in feasible and usable code.	
702	Software Design Review Software design review shall be held at the completion of the software design documentation. Review shall meet the design verification requirements of Requirement 3 of Part I. Review shall evaluate the technical adequacy of the design approach, and assure internal completeness, consistency, clarity, and correctness of the software design, and shall verify that the software design is traceable to requirements.	
703	Development Documentation Review After completion of the testing phase (and the installation phase if necessary) the development cycle documentation shall be reviewed ad approved to assure completion and acceptability.	
800	Problem Reporting & Corrective Action A formal procedure of software problem reporting and corrective action shall be established for software errors and failures. This problem reporting system shall assure that problems are promptly reported to affected organizations to assure formal processing of problem resolutions. Problems in software may be classified by the organization responsible for the evaluation. Any classification system shall have defined criteria based on the impact of the software output. Corrective action by the responsible organization shall assure that: a) problems are identified, evaluated, documented, and if required, corrected; b)problems are assessed for impact on past and present applications of the software by the responsible organization; c) corrections or changes shall be controlled in accordance with paragraph 502 of	204/Problem Reporting and Corrective Action (Part 1, Requirement 16) a) Methods for documenting, evaluating, and correcting software problems shall: 1- describe the evaluation process for determining whether a reported problem is an error or other type of problem; 2- define responsibilities for disposition of the problem reports, including notification to the originator of the results of the evaluation. b) When problem is determined to be error, the method shall provide, as appropriate, for: 1- how error relates to appropriate software engineering elements; 2 – how error impacts past and present use of the computer program; 3 – how the corrective action impacts previous development activities; 4 – how users are notified of identified error, its impact and how to avoid the error, pending implementation of corrective actions. Problem reporting and corrective action process shall address the appropriate requirements of Part I, Requirement

	Subpart 2.7; d)preventive actions and corrective actions results are provided to affected organizations.	16.
900	Access Control To the extent appropriate, controls shall be established to permit authorized access and prevent unauthorized access to a computer system.	-
1000	Procurement	300/Software Acquisition Includes software or software services procured in accordance with Part I, or otherwise acquired for use in activities with the scope of Part I.
1001	Software Individuals or organizations developing and supplying software shall be required to have policies and procedures that meet the applicable requirements of Subpart 2.7 as specified in procurement documents. Documentation that is required that is required by this Subpart shall be delivered or made available by the Supplier to the Purchaser. The applicable requirements of the subpart shall become the responsibility of the Purchaser upon receipt of the software. Typically this software enters the Purchaser's organization at the start of the installation and checkout phase. The Supplier shall report software errors, or failures, to the Purchaser, and the Purchaser shall report software errors to the Supplier.	301/Procured Software and Software Services (Part 1, Requirements 4 and 7) Part I, Requirements 4 and 7 for items and services shall be applied to the procurement of software and software services. The Purchaser shall be responsible for the appropriate requirements of Subpart 2.7 upon acceptance of the software or related item (e.g., programmable device). Procurement documents shall identify requirements for Supplier's reporting of software errors to the Purchaser and as appropriate, the purchaser's reporting of software errors to the Supplier. -
1002	Software Services Organization providing software services, such as verification and validation, shall have a plan(s) for software quality assurance that meets the requirements of this Subpart as specified in procurement documents. The user organization shall determine the adequacy of this plan.	
1100	Software Developed Not Using This Subpart Existing software and procured or otherwise acquired software that has not been previously approved under a program consistent with NQA-1 for use in its intended application shall be evaluated in accordance with the requirements of Subpart 2.7. This software shall be uniquely identified and controlled prior to evaluation, and placed under configuration control prior to use as software approved in accordance with Subpart 2.7. The user organization shall perform and document the above evaluation of the software to: a) determine the adequacy to support operation and maintenance, and b) identify the activities to be performed and the documentation that is needed. This determination shall be documented and shall identify as a minimum: 1) capabilities and limitations for intended use; 2) test plans and test cases required to validate the capabilities within the limitations; and 3) instructions for use within the limits of the capabilities. Exceptions from the documentation requirements of Subpart 2.7 and the justification for acceptance shall be documented. The results of the above evaluation and the performance of the activities identified by this evaluation shall be reviewed and approved. The resulting documentation and associated computer program(s) shall establish the current baseline. Revisions to previously baselined software received from organizations not required to follow Subpart 2.7 shall be evaluated according to criteria of this Section.	302/Otherwise Acquired Software Software that has not been previously approved under a program consistent with this Standard for use in its intended application (e.g. freeware, shareware, procured commercial off-the-shelf, or otherwise acquired software), shall be evaluated in accordance with the requirements of Subpart 2.7. The software shall be identified and controlled prior to evaluation. The evaluation, specified in this section, shall be performed and documented to determine adequacy to support operation and maintenance and identify the activities to be performed and the documentation that is needed. This determination shall be documented and shall identify as a minimum: a – capabilities and limitations for intended use; b – test plans and test cases required to demonstrate the capabilities within the limitations; and c – instructions for use within the limits of the capabilities. Exceptions from the documentation requirements of Subpart 2.7 and the justification for acceptance shall be documented. The results of the above evaluation and the performance of the actions necessary to accept the software, shall be reviewed and approved. The resulting documentation and associated computer program(s) shall establish the current baseline. Revisions to previously baseline software received from organizations not required to follow Subpart 2.7 shall be evaluated in accordance with Subpart 2.7.

Table C-1. Comparison of Subpart 2.7 from NQA-1-1997 with NQA-1a-1999 and NQA-1-2000 (Continued)

Section	NQA-1-1997; Subpart 2.7	NQA-1A-1999 AND NQA-1-2000; SUBPART 2.7
1200	Records Record copies of required documentation shall be retained with other project records as required by codes, standards, specifications, plans, or procedures.	Part I, 900/Documentation and Records
1300	References These standards were used: • ANSI/IEEE 729, Glossary of Software Engineering Terminology • ANSI/IEEE 1012, Software Verification and Validation Plans.	700/References These standards were used: • IEEE Std. 7-4.3.2-1993, IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations • ANSI/IEEE Std. 610.12-1990, Glossary of Software Engineering Terminology.
-	-	600/Support Software Support software includes software tools and system software. As appropriate, the software engineering method, software acquisition method, or both shall establish the need for software tools.
-	-	601/Software Tools Software tools shall be evaluated, reviewed, tested, and accepted for use, and placed under configuration control as part of the software development cycle of a new or revised software product. Software tools that do not affect the performance of the software need not be placed under configuration control. In cases involving modifications of software products using the software tools, the configuration of the support software associated with that modification shall be managed. Changes to the software tool shall be evaluated for impact on the software product to determine the level of reviews and retesting that will be required.
		602/System Software System software consists of the online computer programs used to provide basis or general functionality and facilitate the operation and maintenance of the application computer program. Examples include: lower level software layers, assemblers, interpreters, diagnostics, and utilities. System software shall be evaluated, reviewed, tested, and accepted for use as part of the software development cycle of a new or revised software product. System software shall be placed under configuration change control. Changes to the system software shall be evaluated for impact on the software product to determine the level of reviews and retesting that will be required.

Table C-2. NQA-1-2000, Part I Contents

	Section/Purpose
	Introduction
100	Purpose
200	Applicability
300	Responsibility
400	Terms and Definitions
	Requirements
1	Organization
2	Quality Assurance Program
3	Design Control
4	Procurement Document Control
5	Instructions, Procedures, and Drawings
6	Document Control
7	Control of Purchased Items and Services
8	Identification and Control of Items
9	Control of Special Processes
10	Inspection
11	Test Control
12	Control of Measuring and Test Equipment
13	Handling, Storage, and Shipping
14	Inspection, Test, and Operating Status
15	Control of Nonconforming Items
16	Corrective Action
17	Quality Assurance Records
18	Audits

Table C-3. NQA-1-2000, Part I, Requirement 3

Paragraph	Content
100	Basic
200	Design Input
300	Design Process
400	Design Analyses
401	Use of Computer Programs
402	Documentation of Design Analyses
500	Design Verification
501	Methods (501.1 Design Reviews; 501.2 Alternated Calculations; 501.3 Qualification Tests)
600	Change Control
601	Configuration Management of Operating Facilities
700	Interface Control
800	Software Design Control
801	Software Design Process
801.1	Identification of Software Design Requirements
801.2	Software Design
801.3	Implementation of the Software Design
801.4	Software Design Verification
801.5	Computer Program Testing
802	Software Configuration Management
802.1	Configuration Identification
802.2	Configuration Change Control
802.3	Configuration Status Control
900	Documentation and Records

Table C-4. NQA-1-2000, Part I, Requirement 11

Paragraph	Content
100	Basic
200	Test Requirements
300	Test Procedures (Other Than for Computer Programs)
400	Computer Program Test Procedures
500	Test Results
600	Test Records

Appendix D. Applicable Consensus Standards and Guides

1.	American National Standard, *Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry*, ANSI/ANS-10.4-1987 (Currently under revision).
2.	IEEE Standard 610.12, *IEEE Standard Glossary of Software Engineering Terminology*.
3.	IEEE Standard 730, *IEEE Standard for Software Quality Assurance Plans*.
4.	IEEE Standard 730.1, *IEEE Guide for Software Quality Assurance Planning*.
5.	IEEE Standard 828, *IEEE Standard for Software Configuration Management Plans*.
6.	IEEE Standard 829, *IEEE Standard for Software Test Documentation*.
7.	IEEE Standard 830, *Software Requirements Specifications*
8.	IEEE Standard 1008, *Software Unit Testing*
9.	IEEE Standard 1012, *IEEE Standard for Software Verification and Validation;* IEEE Standard 1012a, *IEEE Standard for Software Verification and Validation – Supplement to 1012*
10.	IEEE Standard 1016.1, *IEEE Guide for Software Design Descriptions* IEEE Standard 1016-1998, *IEEE Recommended Practice for Software Design Descriptions*
11.	IEEE Standard 1063, *IEEE Standard for Software User Documentation*
12.	IEEE Standard 1074, *IEEE Standard for Developing Software Life Cycle Processes*
13.	IEEE/EIA Standard 12207.0*, Industry Implementation of International Standard ISO/IEC 12207 Standard for Information Technology – Software Life Cycle Processes*
14.	IEEE/EIA Standard 12207.1, *Industry Implementation of International Standard ISO.IEC 12207 Standard for Information Technology – Software Life Cycle Processes – Life Cycle Data*
15.	IEEE/EIA Standard 12207.2, *Industry Implementation of International Standard ISO.IEC 12207 Standard for Information Technology – Software Life Cycle Processes – Implementation Considerations*

Appendix E. Application to the MACCS2 Computer Code

An estimate of the program and level of effort required to upgrade the MACCS2 computer software was prepared by SNL using NP-19 (Bixler, 2000). NP-19 was identified earlier, and is a SNL procedural guide that implements an earlier version of Subpart 2.7 to NQA-1, specifically NQA-2a-1990. The minimum set of actions, to be applied to MACCS2 are taken from Bixler (2000) and are as follows:

- Create a Primitive Baseline (PB) document to establish the SQA status of the existing code
- Write a Software Requirements Document (SRD)
- Establish a Verification and Validation Plan (VVP) based on the SRD
- Create an Implementation Document (ID) to describe the process of generating the executable software modules
- Update, the User's Manual (UM)
- Generate a Validation Document (VD), to measure the performance of the software against the criteria specified in the VVP
- Perform Installation and Checkout (I&C) to verify correct installation on all supported platforms
- Implement a Software Configuration Control System (CC)
- Implement a Software Problem Reporting System (SPR).

The overall SQA upgrade program was estimated to require 1.5 full-time equivalent years to complete, and is matched against the requirements discussed in this document (see Table 5) in Table 8. This compared favorably with an independent 2-FTE-year estimate generated by East, but follows ANSI/ANS-10.4 (WSRC, 1998).

The SQA evaluation performed for MACCS2 would follow a similar plan but would be updated to follow the primary criteria and implementation procedures discussed in this report.

Table 8. Comparison of SQA Upgrade Steps Discussed in Bixler (2000) with WSRC Quality Assurance Plan 20-1

ASME NQA-1-2000 requirements	SNL NP 19-1	Level B Existing Software
Software Classification		X
SQA Procedures/Plans		X
Dedication		
Evaluation	PB	X
Requirements	SRD	X
Design		
Implementation		
Testing	VVP, VD	X
User Instructions	ID, UM	X
Acceptance Test	I&C	
Operation and Maintenance		X
Configuration Control	CC	X
Error Impact	SPR	X
Access Control		X

It is concluded that each of the six designated safety analysis codes falls in the *existing* code category, and is classified as "B" software by application. The latter are software applications

- Whose failure to properly function may have an indirect effect on nuclear safety protection systems or toxic materials hazard systems, that are used to keep nuclear or toxic material hazard exposure to the general public and workers below regulatory or evaluation guidelines,

or

Whose results are used to make decisions that could result in death or serious injury or are part of the evaluation in accident analyses.