Defense Nuclear Facilities Safety Board Recommendation 2002-1
Software Quality Assurance Improvement Plan
Commitment 4.2.1.3:

Software Quality Assurance Improvement Plan: EPIcode Gap Analysis

Final Report

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

May 2004

INTENTIONALLY BLANK

FOREWORD

This report documents the outcome of an evaluation of the Software Quality Assurance (SQA) attributes of the chemical source term and atmospheric dispersion computer code, EPIcode, relative to established requirements. This evaluation, a "gap analysis", is performed to meet commitment 4.2.1.3 of the Department of Energy's Implementation Plan to resolve SQA issues identified in the Defense Nuclear Facilities Safety Board Recommendation 2002-1.

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

Chip Lagdon
EH-31/GTN
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	1. Interim Report	1. Original Issue
2. Entire Document	2. Final Report, May 3, 2004	2. Updated all sections per review comments.

INTENTIONALLY BLANK

CONTENTS

TABLES

INTENTIONALLY BLANK

FIGURES

None

INTENTIONALLY BLANK

Software Quality Assurance Improvement Plan: EPIcode Gap Analysis

EXECUTIVE SUMMARY

The Defense Nuclear Facilities Safety Board (DNFSB) issued Recommendation 2002-1 on *Quality Assurance for Safety-Related Software* in September 2002 (DNFSB 2002). The Recommendation identified a number of quality assurance issues for software used in the Department of Energy (DOE) facilities for analyzing hazards, and designing and operating controls that prevent or mitigate potential accidents. The development and maintenance of a collection, or "toolbox," of high-use, Software Quality Assurance (SQA)-compliant safety analysis codes is one of the major improvement actions discussed in the *Implementation Plan for Recommendation 2002-1 on Quality Assurance for Safety Software at Department of Energy Nuclear Facilities*. A DOE safety analysis toolbox would contain a set of appropriately quality-assured, configuration-controlled, safety analysis codes, managed and maintained for DOE-broad safety basis applications.

The EPIcode 7.0 software for chemical source term and atmospheric dispersion and consequence analysis, is one of the codes designated for the toolbox. To determine the actions needed to bring the EPIcode 7.0 software into compliance with the SQA qualification criteria, and develop an estimate of the resources required to perform the upgrade, the Implementation Plan has committed to sponsoring a code-specific gap analysis document. The gap analysis evaluates the software quality assurance attributes of EPIcode 7.0 against identified criteria.

The balance of this document provides the outcome of the EPIcode gap analysis compliant with NQA-1-based requirements. Of the ten SQA requirements for existing software at the Level B classification (important for safety analysis but whose output is not applied without further review), two requirements are met at acceptable level, i.e., *Classification* (1) and *User Instructions* (7). Improvement actions are recommended for EPIcode to fully meet the remaining eight requirements. This evaluation outcome is deemed acceptable because: (1) EPIcode is used as a tool, and as such its output is applied in safety analysis only after appropriate technical review; (2) User-specified inputs are chosen at a reasonably conservative level of confidence; and (3) Use of EPIcode is limited to those analytic applications for which the software is intended.

Suggested remedial actions for this software would warrant upgrading software documents. The complete list of revised 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 Manual.

It is estimated that a concentrated program to upgrade the SQA pedigree of EPIcode to be compliant with the ten criteria discussed here would require fourteen to sixteen full-time equivalent (FTE)-months.

It was determined that the EPIcode 7.0 does meet its intended function for use in supporting documented safety analysis. However, as with all safety-related software, users should be aware of current limitations and capabilities of the software for supporting safety analysis. Informed use of the code can be assisted by appropriate use of current EPIcode documentation and the EPIcode guidance report for DOE safety analysts, *EPIcode Computer Code Application Guidance for Documented Safety Analysis*, (DOE, 2004). Furthermore, while SQA improvement actions are recommended for EPIcode, no evidence has been found of programming, logic, or other types of software errors in EPIcode 7.0 that have led to non-conservatisms in nuclear facility operations, or in the identification of facility controls.

INTENTIONALLY BLANK

1.0 Introduction

This document reports on the results of a gap analysis for Version 7.0 of the EPIcode computer code. The intent of the gap analysis is to determine the actions needed to bring the designated software into compliance with established Software Quality Assurance (SQA) criteria. A secondary aspect of this report is to develop an estimate of the level of effort required to upgrade each code based on the gap analysis results.

1.1 Background: Overview of Designated Toolbox Software in the Context of 10 CFR 830

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.

While progress was made in resolving several of the issues raised in TECH-25, the DNFSB issued Recommendation 2002-1 on *Quality Assurance for Safety-Related Software* in September 2002. 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. The Recommendation identified a number of quality assurance issues for software used in the DOE facilities for analyzing hazards, and designing and operating controls that prevent or mitigate potential accidents. The development and maintenance of a collection, or "toolbox," of high-use, SQA-compliant safety analysis codes is one of the major commitments contained in the February 28, 2003 *Implementation Plan for Recommendation 2002-1 on Quality Assurance for Safety Software at Department of Energy Nuclear Facilities* (IP). In time, the DOE safety analysis toolbox will contain a set of appropriately quality-assured, configuration-controlled, safety analysis codes, managed and maintained for DOE-broad safety basis applications.

Six computer codes, including ALOHA (chemical release dispersion/consequence analysis), CFAST (fire analysis), EPIcode (chemical release dispersion/consequence analysis), GENII (radiological dispersion/consequence analysis), MACCS2 (radiological dispersion/consequence analysis), and MELCOR (leak path factor analysis), were designated by DOE for the toolbox (DOE/EH, 2003). It is found that this software provides generally recognized and acceptable approaches for modeling source term and consequence phenomenology, and can be applied as appropriate to support accident analysis in Documented Safety Analyses (DSAs).

As one of the designated toolbox codes, EPIcode Version 7.0 is likely to require some degree of quality assurance improvement before meeting current SQA standards. The analysis of this document evaluates EPIcode Version 7.0 relative to current software quality assurance criteria. It assesses the extent of the deficiencies, or gaps, to provide DOE and the software developer the extent to which minimum upgrades are needed. The overall assessment is therefore termed a "gap" analysis.

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.

Ideally, each toolbox code owner will provide complete information on the SQA programs, processes, and procedures used to develop their software. However, the gap analysis itself will be performed by a SQA evaluator. 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 provides key information to DOE, code developers, and code users.

DOE obtains 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 is provided:

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

DOE safety analysts and code users benefit from:

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

1.4 Scope

This analysis is applicable to the EPIcode, one of the six designated toolbox codes for safety analysis. While EPIcode is the subject of the current report, other safety analysis software considered for the toolbox in the future may be evaluated with the same process applied here. The template outlined here is applicable for any analytical software as long as the primary criteria are ASME NQA-1, 10 CFR 830, and related DOE directives discussed in DOE (2003e).

1.5 Purpose

The purpose of this report is to document the gap analysis performed on the EPIcode as part of DOE's implementation plan on SQA improvements.

1.6 Methodology for Gap Analysis

The gap analysis for EPIcode is based on the plan and criteria described in *Software Quality Assurance Plan and Criteria for the Safety Analysis Toolbox Codes* (DOE 2003e). The overall methodology for the gap analysis is summarized in Table 1-1. The gap analysis reported here utilizes ten of the fourteen topical areas listed in DOE (2003e) related to software quality assurance to assess the quality of the EPIcode 7.0 computer code. The ten areas are those particularly applicable to the software development, specifically: (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, and (13) Error Impact. Each area, or requirement, is assessed individually in Section 4. Each area or requirement is assessed individually in Section 4.

Requirements 3 (Dedication), 4 (Evaluation), 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 evaluating the areas listed above. Requirement 11 (Operation and Maintenance) is only partially applicable to software development, and is interpreted to be applicable mostly to the software user organization.

An information template was transmitted to the Safety Analysis Software Developers on 20 October 2003 to provide basic information as input to the gap analysis process (O'Kula, 2003). The core section of the template is attached as Appendix A to the present report. It is noted that the written response provided by the EPIcode software developer to the information template was incomplete.

Table 1-1 — Plan for SQA Evaluation of Existing Safety Analysis Software[1]

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).

[1] Originally documented as Table 2-2 in DOE (2003e).

Phase	Procedure
	c. Identify documents that are acceptable from SQA perspective. Note inadequate documents as appropriate.

Table 1-1 — Plan for SQA Evaluation of Existing Safety Analysis Software (continued)

Phase	Procedure
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.
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.

1.7 Summary Description of Software Being Reviewed

The gap analysis was performed on version 7.0 of the EPIcode® (note: EPIcode® is a registered trademark of Homann Associates, Inc.). EPIcode was developed by Homann Associates, Inc., which maintains and upgrades the code. The code is commercially available from Homann Associates, Inc. The technical contact for EPIcode is the code author, Steven Homann (www.epicode.com, or epicode@aol.com).

EPIcode performs calculations for source terms and downwind concentrations. Source term calculations determine the rate at which the chemical material is released to the atmosphere, release height, release duration, and the form and properties of the chemical upon release. The analyst specifies the chemical and then either specifies the chemical source term rate or provides EPIcode with the necessary information and data to calculate a steady evaporation rate when the scenario involves a spill of a chemical liquid. Releases may be elevated either through discharge from a stack or as a result of plume rise from buoyancy or momentum effects. The EPIcode considers the chemical cloud emission to be neutrally buoyant and applies standard Gaussian puff and plume models as appropriate. In addition to the source term and downwind concentration calculations, EPIcode supports the use of concentration limits

for the purpose of consequence assessment (e.g., assessment of human health risks from contaminant plume exposure). When available, data for Immediately Dangerous to Life or Health (IDLH), Emergency Response Planning Guidelines (ERPGs), Department of Energy Temporary Emergency Exposure Limits (TEELs), and EPA Acute Exposure Guideline Limits (AEGLs) have been incorporated into the chemical library of EPIcode.

A brief summary of EPIcode that was supplied code developer is summarized in Table 1-2.

Table 1-2 — Summary Description of EPIcode Software

Type	**Specific Information**
Code Name	EPIcode®
Version of the Code	Version 7.0
Developing Organization and Sponsor Information	Homann Associates, Inc.
Auxiliary Codes	N/A
Software Platform/Portability	Microsoft™ Visual Basic Professional 6.0, PC-based
Coding and Computer(s)	Microsoft™ Visual Basic Professional 6.0, PC-based 80486 or Pentium processor Windows 95/98/00/NT/XP OS
Technical Support Point of Contact	Homann Associates, Inc. (510) 490-6379 epicode@aol.com www.epicode.com
Code Procurement Point of Contact	Homann Associates, Inc. (510) 490-6379 epicode@aol.com www.epicode.com
Code Package Label/Title	EPIcode 7.0, single CD
Contributing Organization(s)	N/A
Recommended Documentation - Supplied with Code Transmittal upon Distribution or Otherwise Available	EPIcode documentation and user manual are components of EPIcode 7.0 onboard runtime library. Users access this information via a command button or the F1 key.

Table 1-2 — Summary Description of EPIcode Software (Continued)

Type	Specific Information
Input Data/Parameter Requirements	Source Term substance: via name, CAS number, DOT Number, TEEL database name (rev 19). Source Term: Total release rate or total release (g/s, g, etc.) Airborne Fraction (AF) .The fraction of the total quantity of material that remains airborne. Deposition velocity (cm/sec). Effective release height (m). Explosive Release Modules: High Explosive (pounds TNT equivalent). Fuel Fire Module: Volume of Fuel (gallons), Burn duration (minutes), Heat emission rate (calories/second). Radius of fire zone (m). Optional Source Term Geometry: Horizontal Dimension (meters), Vertical Dimension (meters), Height (meters). Wind Speed (m/s) at input reference height. Wind Direction (compass degrees) for geographical mapping overlay Stability Class (A-G) Receptor Height (meters). Inversion Layer Height (meters) Washout Coefficient (1/second), for washout plume depletion and ground deposition.
Summary of Output	Results from EPIcode atmospheric release calculations can be displayed or printed in tabular form or as graphic plots showing the downwind centerline concentration or concentration contours. All files can be archived. EPIcode contours can also be displayed on any .bmp image, e.g., satellite maps, map photos, etc. Off-axis locations can also be included in the tabular output.
Nature of Problem Addressed by Software	EPIcode has been specially developed to provide emergency response personnel, emergency planners, and health and safety professionals with a software tool to aid them in evaluating the atmospheric release of toxic substances.

Table 1-2 — Summary Description of EPIcode Software (Continued)

Type	Specific Information
Significant Strengths of Software	EPIcode is completely menu-driven and easy to use. EPIcode uses the same algorithms and methodologies outlined in EPA document titled "Technical Guidance for Hazards Analysis -Emergency Planning for Extremely Hazardous Substances," U.S. Environmental Protection Agency, Federal Emergency Management Agency, and U.S. Department of Transportation, December 1987. EPIcode output always contains all of the input assumptions, and the calculated radii of the vulnerable zones are in exact agreement with the above EPA document. EPIcode contains a library of over 2,000 chemical substances along with the associated exposure levels accepted by various professional organizations and regulatory agencies. These include all of the current American Industrial Hygiene Association Emergency Response Planning Guidelines (ERPGs), Department of Energy Temporary Emergency Exposure Limits (TEELs), and EPA Acute Exposure Guideline Limits (AEGLs). The EPIcode Library also contains information on substances listed in the Threshold Limit Values for Chemical Substances and Physical Agents and Biological Exposure Indices published by the American Conference of Governmental Industrial Hygienists. IDLH (Immediately Dangerous to Life or Health) data are also included when available. Virtual source terms are used to more accurately model the initial distribution of material associated with explosions or fires.
Known Restrictions or Limitations	The atmospheric model included in the code does not model the impact of terrain effects on atmospheric dispersion. A single wind direction and input height is assumed.
Preprocessing (set-up) time for Typical Safety Analysis Calculation	Few minutes or less
Execution Time	Less than 5 seconds
Computer Hardware Requirements	Any PC running Microsoft™ Windows 95/98/00/NT/XP OS (Fully operational on Apple™ computers running Windows 95/98 emulator software)
Computer Software Requirements	Microsoft™ Windows 95/98/00/NT/XP OS
Other Versions Available	N/A

Table 1-2 — Summary Description of EPIcode Software (Continued)

Type	Specific Information
Individual(s) completing this information form: Name: Organization: Telephone: Email: Fax:	Steven Homann Homann Associates, Inc. Voice: (510) 490-6379 Email: epicode@aol.com Fax: (510) 490-6379 Web: www.epicode.com

The set of documents reviewed as part of the gap analysis are listed in Table 1-3.

Table 1-3 — Software Documentation Reviewed for EPIcode

No.	Reference
1.	*EPIcode Version 7.0 User Documentation (*EPIcode, 2003) {Online Help distributed with software package}
2.	*Technical Guidance for Hazards Analysis: Emergency Planning for Extremely Hazardous Substances* (EPA, 1987) {Source of algorithms and methodologies that are used in EPIcode}
3.	*Risk Management Program Guidance for Offsite Consequences* (EPA, 1999) {Source of updated evaporation model (use of 0.67 for mass transfer coefficient instead of 0.24) that is cited in Ref. 2 above (EPA, 1987)}
4.	*EPIcode User's Guide, Version 6.0* (Homann, 1996) {User documentation for earlier version, which documents more sample problems than current versions cited in Ref. 1}

2.0 Assessment Summary Results

2.1 Criteria Met

Of the ten general topical quality areas assessed in the gap analysis, two satisfactorily met the criteria. The analysis found that the EPIcode SQA program, in general, met criteria for *Software Classification* and *User Instructions*, Requirements 1 and 7, respectively. The remaining eight topical quality areas were judged either not wholly compliant with the SQA criteria, and/or lacked documentation to confirm compliance. The eight areas that should be addressed for improvement actions are listed in Section 2.2 (Exceptions to Requirements). Details on the evaluation process relative to the requirements and the criteria applied, are found in Section 4.

2.2 Exceptions to Requirements

Exceptions to criteria found for EPIcode 7.0 are listed below in Table 2-1. The requirement is given, the reason the requirement was not met is provided, and action(s) are listed to correct the exceptions. The ten criteria evaluated are those predominantly executed by the software developer. However, it is noted that criteria for SQA Procedures/Plan, Testing, Acceptance Test, Configuration Control, and Error Notification also have requirements for the organization implementing the software. These criteria were assessed in the present evaluation only from the code developer perspective.

Table 2-1 — Summary of Important Exceptions, Reasoning, and Suggested Remediation

No.	Criterion	Reason Not Met	Remedial action(s)
1.	SQA Procedures/Plans (Section 4.2)	SQA Plans and Procedures were not available for the gap analysis.	SQA Plans and Procedures should be developed and made available for review.
2.	Requirements Phase (Section 4.3)	A Software Requirements Document does not exist for review. Thus, it was necessary to infer requirements from draft model description and user guidance documents.	A Software Requirements Document should be prepared and made available for review.
3.	Design Phase (Section 4.4)	A Software Design Document does not exist for review. Thus, it was necessary to infer the intent of the design from draft model description and user guidance documents.	A Software Design Document should be prepared and made available for review.
4	Implementation Phase (Section 4.5)	Documentation to support the implementation is lacking.	A verifiable, written set of SQA plans and procedures including implementation, test case descriptions, and associated criteria related to

No.	Criterion	Reason Not Met	Remedial action(s)
			design should be made available.
5.	Testing Phase (Section 4.6)	A Software Testing Report Document does not exist for review.	A Software Testing Report Document should be prepared and made available for review.
6	Acceptance Test (Section 4.8)	A verifiable, written set of SQA plans and procedures, which would include acceptance testing documentation, is lacking.	Documented acceptance testing should be developed..
7.	Configuration Control (Section 4.9)	A Configuration and Control Document does not exist for review.	A Configuration and Control Document should be prepared and made available for review.
8.	Error Notification (Section 4.10)	An Error Notification and Corrective Action Report do not exist for review.	While a Software Problem Reporting system is apparently in place, written documentation should be provided to the Central Registry for verification of its effectiveness.

2.3 Areas Needing Improvement

The gap analysis identified a few improvements that could be made related to the code.. The recommended upgrades are listed in Table 2-2. These recommended upgrades for EPIcode focus on adding technical capabilities to broaden the use of EPIcode for DSA-type applications and reducing conservatism in the results.

Table 2-2 — Summary of Important Recommendations for EPIcode

No.	Recommendation
1.	Add capability to model dense gas behavior or provide a warning when the release scenario has conditions that might lead to dense gas type of atmospheric transport and dispersion.
2.	Add capability to read from a file of hourly meteorological data over a one-year period, calculate consequences for each hourly entry, and output the 50^{th} and 95^{th} percentile results.
3.	Add capability to use surface roughness input to adjust the rural vertical dispersion coefficient when the input value is greater than 3 cm and less than 100 cm.

2.4 Conclusion Regarding Codes Ability to Meet Intended Function

The EPIcode 7.0 software was evaluated to determine if the software in its current state meets the intended function in a safety analysis context as assessed in this gap analysis. When the code is run for

the intended applications as detailed in the code guidance document, EPIcode *Computer Code Application Guidance for Documented Safety Analysis*, (DOE 2004), it is judged that it will meet its intended function.

3.0 Lessons Learned

Additional opportunities and venues should be sought for training and user qualification on safety analysis software. This is a long-term recommendation for EPIcode and other designated software for the DOE toolbox.

4.0 Detailed Results of the Assessment Process

Ten topical areas or requirements are presented in the assessment as listed in Table 4.0-1.
In the tables that follow, criteria and recommendations are labeled as (1.x, 2.x, …10.x) with the first value (1., 2., …) corresponding to the topical area and the second value (x), the sequential table order.

Table 4-0-1— Cross-Reference of Requirements with Subsection and Entry from DOE (2003e)

Subsection (This Report)	Corresponding Entry Table 3-3 from DOE (2003e)	Requirement	ASME NQA-1 2000 Section/Consensus Standards
4.1	1	Software Classification	ASME NQA-1 2000 Section 200
4.2	2	SQA Procedures/Plans	ASME NQA-1 2000 Section 200; IEEE Std. 730, *IEEE Standard for Software Quality Assurance Plans*
4.3	5	Requirements Phase	ASME NQA-1 2000 Section 401; IEEE Standard 830, *Software Requirements Specifications*
4.4	6	Design Phase	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*
4.5	7	Implementation Phase	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*
4.6	8	Testing Phase	ASME NQA-1 2000 Section 404; IEEE Std. 829, *IEEE Standard for Software Test Documentation;* IEEE Standard 1008, *Software Unit*

			Testing
4.7	9	User Instructions	ASME NQA-1 2000 Section 203; IEEE Standard 1063, *IEEE Standard for Software User Documentation*
4.8	10	Acceptance Test	ASME NQA-1 2000 Section 404; IEEE Std. 829, *IEEE Standard for Software Test Documentation;* IEEE Standard 1008, *Software Unit Testing*
4.9	12	Configuration Control	ASME NQA-1 2000 Section 405; ASME NQA-1 2000 Section 406
4.10	13	Error Notification	ASME NQA-1 2000 Section 203

4.1 Topical Area 1 Assessment: Software Classification

This area corresponds to the requirement entitled Software Classification in Table 3-3 of (DOE 2003e).

4.1.1 Criterion Specification and Result

Table 4.1-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

Sufficient documentation is provided with software transmittal to make an informed determination of the classification of the software. A user of the EPIcode software for safety analysis applications would be expected to interpret the information on the software in light of the requirements for atmospheric dispersion and consequence analysis discussed in Appendix A to DOE-STD-3009-94 to decide on an appropriate safety classification. For most organizations, the safety class or safety significant classification, or Level B in the classification hierarchy discussed in DOE (2003e), would be selected.

Table 4.1-1 — Subset of Criteria for Software Classification Topic and Results

Criterion Number	Criterion Specification	Compliant	Summary Remarks
1.1	The code developer must provide sufficient information to allow the user to make an informed decision on the classification of the software.	Yes.	It is concluded that sufficient information is provided with the documentation that is transmitted with the software for the user to make an informed determination of the classification of the software. For most DSA applications, the safety class or safety significant classification, or Level B in the classification hierarchy discussed in DOE (2003e), would be selected,

Criterion Number	Criterion Specification	Compliant	Summary Remarks
			which by definition relate to 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.

4.1.2 Sources and Method of Review

Documentation supplied or referenced with the software package and the software developer's partial response to the software information template shown in Appendix A were used as the basis for response to this requirement.

4.1.3 Software Quality-Related Issues or Concerns

There are no SQA issues or concerns relative to this requirement.

4.1.4 Recommendations

No recommendations are provided at this time.

4.2 Topical Area 2 Assessment: SQA Procedures and Plans

This area corresponds to the requirement entitled SQA Procedures and Plans in Table 3-3 of (DOE 2003e).

From the limited information received from the software developer, formal, published SQA procedures and plans were not developed. While it is possible that most elements of a compliant SQA program were

followed in the development of EPIcode 7.0, the lack of written documentation prevents an independent evaluator from making a definitive confirmation.

4.2.1 Criterion Specification and Result

Table 4.2-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

Table 4.2-1 — Subset of Criteria for SQA Procedures and Plans Topic and Results

Criterion Number	Criterion Specification	Compliant	Summary Remarks
2.1	Procedures/plans for SQA (**SQA Plan**) have identified organizations responsible for performing work; independent reviews, etc.	No.	It is recommended that a SQA plan be developed to provide a framework for configuration control, code maintenance, and support of future upgrades.
2.2	Procedures/plans for SQA (**SQA Plan**) have identified software engineering methods.	No.	See Criterion 2.1 summary remarks.
2.3	Procedures/plans for SQA (**SQA Plan**) have identified documentation to be required as part of program.	No.	See Criterion 2.1 summary remarks.
2.4	Procedures/plans for SQA (**SQA Plan**) have identified standards, conventions, techniques, and/or methodologies which shall be used to guide the software development, methods to ensure compliance with the same.	No.	See Criterion 2.1 summary remarks.
2.5	Procedures/plans for SQA (**SQA Plan**) have identified software reviews and schedule.	No.	See Criterion 2.1 summary remarks.
2.6	Procedures/plans for SQA (**SQA Plan**) have identified methods for error reporting and corrective actions.	No.	See Criterion 2.1 summary remarks.

4.2.2 Sources and Method of Review

Documentation supplied or referenced with the software package and the software developer's partial response to the software information template shown in Appendix A were used as the basis for response to this requirement.

4.2.3 Software Quality-Related Issues or Concerns

Lack of a verifiable, written set of SQA plans and procedures for EPIcode should be addressed.

4.2.4 Recommendations

Recommendations related to this topical area are provided as follows:

- It is recommended that a SQA plan be developed to provide a framework for configuration control, code maintenance, and support of future upgrades.

4.3 Topical Area 3 Assessment: Requirements Phase

This area corresponds to the requirement entitled Requirements Phase in Table 3-3 of (DOE 2003e).

4.3.1 Criterion Specification and Result

Table 4.3-1 lists the subset of criteria reviewed for this topical area and summarizes the findings

Table 4.3-1 — Subset of Criteria for Requirements Phase Topic and Results

Criterion Number	Criterion Specification	Compliant	Summary Remarks
3.1	Software requirements for the subject software have been established.	Yes.	Implicitly fulfilled. The EPIcode program was developed to provide emergency response personnel and emergency planners with a software tool to evaluate downwind concentrations from the atmospheric release of toxic substances. Specifically, the online user's documentation states that EPIcode was designed to produce calculated radii of the vulnerable zones that are in exact agreement with the EPA document, "Technical Guidance for Hazards Analysis -Emergency Planning for Extremely Hazardous Substances" (EPA, 1987).
3.2	Software requirements are specified, documented, reviewed and approved.	No.	A verifiable, written set of SQA plans and procedures, which would include software requirements, is lacking for EPIcode.
3.3	Requirements define the functions to be performed by the software and provide detail and information necessary to design the software.	Yes.	EPIcode strictly follows the well-established Gaussian model. EPIcode uses no "black-box" techniques. All algorithms are presented and fully referenced in the onboard Software User Documentation. EPIcode uses the same algorithms and methodologies outlined in

Criterion Number	Criterion Specification	Compliant	Summary Remarks
			EPA document titled "Technical Guidance for Hazards Analysis - Emergency Planning for Extremely Hazardous Substances," U.S. Environmental Protection Agency, Federal Emergency Management Agency, and U.S. Department of Transportation, December 1987.
3.4	A **Software Requirements Document**, or equivalent defines requirements for functionality, performance, design inputs, design constraints, installation considerations, operating systems (if applicable), and external interfaces necessary to design the software.	Partial.	As stated above, the online user's documentation implicitly states requirements. The user's documentation also addresses, at least partially, installation, operating systems and design inputs.
3.5	Acceptance criteria are established in the software requirements documentation for each of the identified requirements.	Partial.	According to the online user's documentation, "EPIcode output always contains all of the input assumptions, and the calculated radii of the vulnerable zones are in exact agreement with the EPA document. This demonstrates correct implementation of the basic Gaussian algorithms contained in the EPA document."

Additional Detail The Gaussian model is the basic workhorse for atmospheric dispersion calculations and has found its way into most governmental guidebooks. The Gaussian model has also been used and accepted by the Environmental Protection Agency (EPA, 1978). The adequacy of this model for making initial dispersion estimates or worst-case safety analyses has been tested and verified for many years.

4.3.2 Sources and Method of Review

Documentation supplied or referenced with the software package and the software developer's partial response to the software information template shown in Appendix A were used as the basis for response to this requirement.

4.3.3 Software Quality-Related Issues or Concerns

Lack of a verifiable, written set of SQA plans and procedures, which would include written software requirements, for EPIcode should be addressed.

4.3.4 Recommendations

Recommendations related to this topical area are provided as follows:

- Documented software requirements will be needed for EPIcode to meet all prerequisites for the DOE toolbox.

4.4 Topical Area 4 Assessment: Design Phase

This area corresponds to the requirement entitled Design Phase in Table 3-3 of (DOE 2003e).

4.4.1 Criterion Specification and Result

Table 4.4-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

Table 4.4-1 — Subset of Criteria for Design Phase Topic and Results

Criterion Number	Criterion Specification	Compliant	Summary Remarks
4.1	The software design was developed, documented, reviewed and controlled.	Uncertain.	Because SQA plans and procedures from the software developer are not available, a thorough evaluation was not possible.
4.2	Code developer(s) 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.	Partial.	Design may be inferred from final software product, but design document was not made available for review.
4.3	The following design should be present and documented: specification of interfaces, overall structure (control and data flow) and the reduction of the overall structure into physical solutions (algorithms, equations, control logic, and data structures).	Uncertain.	See Criterion 4.1 summary remarks.
4.4	The following design should be present and documented: 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.	Uncertain.	See Criterion 4.1 summary remarks.
4.5	The following design should be present and documented: evidence of measures to mitigate the consequences of software design problems. These potential problems include external and internal abnormal conditions and	Not applicable to non-process, instrumentation and control software.	None.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	events that can affect the computer program.		
4.6	A Software Design Document, or equivalent, is available and contains a description of the major components of the software design as they relate to the software requirements.	No.	A verifiable, written set of SQA plans and procedures, which would include software design documentation, is lacking for EPIcode.
4.7	A Software Design Document, or equivalent, is available and contains 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.	No.	See Criterion 4.6 summary remarks.
4.8	A Software Design Document, or equivalent, is available and contains a description of the allowable or prescribed ranges for inputs and outputs.	Yes.	The EPIcode user documentation contains this information.
4.9	A Software Design Document, or equivalent, is available and contains the design described in a manner that can be translated into code.	No.	See Criterion 4.6 summary remarks.
4.10	A Software Design Document, or equivalent, is available and contains 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.	No.	See Criterion 4.6 summary remarks.
4.11	The organization responsible for the design identified and documented the particular verification methods to be used and assured that an Independent Review was performed and documented. This review evaluated the technical adequacy of the design approach; assured internal completeness, consistency, clarity, and correctness of the software design; and verified that the software design is traceable to the requirements.	Uncertain.	While some elements of this criterion may have been met informally, there is no written documentation that allows confirmation.
4.12	The organization responsible for the design assured that the test results adequately demonstrated the	Uncertain.	See Criterion 4.1 summary remarks.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	requirements were met.		
4.13	The Independent Review was performed by competent individual(s) other than those who developed and documented the original design, but who may have been from the same organization.	Uncertain.	While some elements of this criterion may have been met informally, there is no written documentation that allows confirmation.
4.14	The results of the Independent Review are documented with the identification of the verifier indicated.	Uncertain.	See Criterion 4.1 summary remarks.
4.15	If review alone was not adequate to determine if requirements are met, alternate calculations were used, or tests were developed and integrated into the appropriate activities of the software development cycle.	Uncertain.	See Criterion 4.1 summary remarks.
4.16	Software design documentation was completed prior to finalizing the Independent Review.	No.	See Criterion 4.6 summary remarks.
4.17	The extent of the Independent Review and the methods chosen are shown to be a function of: ➢ The importance to safety, ➢ The complexity of the software, ➢ The degree of standardization, and ➢ The similarity with previously proven software.	Uncertain.	See Criterion 4.1 summary remarks.

4.4.2 Sources and Method of Review

Documentation supplied or referenced with the software package and the software developer's partial response to the software information template shown in Appendix A were used as the basis for response to this requirement.

4.4.3 Software Quality-Related Issues or Concerns

Lack of a verifiable, written set of SQA plans and procedures, which would include software design documentation, for EPIcode should be addressed.

4.4.4 Recommendations

Recommendations related to this topical area are provided as follows:

- Documented software design will be needed for EPIcode to meet all prerequisites for the DOE toolbox.

4.5 Topical Area 5 Assessment: Implementation Phase

This area corresponds to the requirement entitled Implementation Phase in Table 3-3 of (DOE 2003e).

4.5.1 Criterion Specification and Result

Table 4.5-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

Table 4.5-1 — Subset of Criteria for Implementation Phase Topic and Results

Criterion Number	Criterion Specification	Compliant	Summary Remarks
5.1	The implementation process resulted in software products such as computer program listings and instructions for computer program use.	Partial.	Elements of this criterion may be inferred from documentation and the final software product, however, the implementation process has not been formally documented..
5.2	Implemented software was analyzed to identify and correct errors.	Uncertain.	Because SQA plans and procedures from the software developer are not available, a thorough evaluation was not possible.
5.3	The source code finalized during verification (this phase) was placed under configuration control.	Uncertain.	See Criterion 5.2 summary remarks.
5.4	Documentation during verification included a copy of the software, test case description and associated criteria that are traceable to the software requirements and design documentation.	No.	A verifiable, written set of SQA plans and procedures, which would include test case descriptions as well as software requirements and design documentation, is lacking for EPIcode.

4.5.2 Sources and Method of Review

Documentation supplied or referenced with the software package and the software developer's partial response to the software information template shown in Appendix A were used as the basis for response to this requirement.

4.5.3 Software Quality-Related Issues or Concerns

Lack of a verifiable, written set of SQA plans and procedures, which would include test case descriptions as well as software requirements and design documentation, for EPIcode should be addressed.

4.5.4 Recommendations

Recommendations related to this topical area are provided as follows:

- A documented implementation process will be needed for EPIcode to meet all prerequisites for the DOE toolbox.

4.6 Topical Area 6 Assessment: Testing Phase

This area corresponds to the requirement entitled Testing Phase in Table 3-3 of (DOE 2003e).

4.6.1 Criterion Specification and Result

Table 4.6-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

Table 4.6-1 — Subset of Criteria for Testing Phase Topic and Results

Criterion Number	Criterion Specification	Compliant	Summary Remarks
6.1	The software was validated by executing test cases.	Yes.	EPIcode uses the same algorithms and methodologies outlined in EPA document titled "Technical Guidance for Hazards Analysis -Emergency Planning for Extremely Hazardous Substances," U.S. Environmental Protection Agency, Federal Emergency Management Agency, and U.S. Department of Transportation, December 1987. According to the code developer, EPIcode output always contains all of the input assumptions, and the calculated radii of the vulnerable zones are in exact agreement with the EPA document. This demonstrates correct implementation of the basic Gaussian algorithms contained in the EPA document.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
6.2	Testing demonstrated 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 provide evidence to ensure that the software adequately and correctly performed all intended functions.	Partial.	The EPIcode user's guide contains 15 example case studies that show how EPIcode can be applied to a wide range of chemical accident scenarios. In nearly half of these examples, the EPIcode results are compared against field measurements or the output of other computer codes. Documentation is lacking, however, to confirm all aspects of this requirement.
6.3	Testing demonstrated that the computer program properly handles abnormal conditions and events as well as credible failures.	Uncertain.	Because SQA plans and procedures from the software developer are not available, a thorough evaluation was not possible.
6.4	Testing demonstrated that the computer program does not perform adverse unintended functions.	Uncertain.	See Criterion 6.3 summary remarks.
6.5	Test Phase activities were performed to assure adherence to requirements, and to assure that the software produces correct results for the test case specified. Acceptable methods for evaluating adequacy of software test case results included: (1) analysis with computer assistance; (2) other validated computer programs; (3) experiments and tests; (4) standard problems with known solutions; (5) confirmed published data and correlations.	Partial.	See Criterion 6.1 summary remarks.
6.6	Test Phase documentation includes test procedures or plans and the results of the execution of test cases. The test results documentation demonstrates successful completion of all test cases or the resolution of unsuccessful test cases and provides direct traceability between the test results and specified software requirements.	No.	A verifiable, written set of SQA plans and procedures, which would include test phase documentation, is lacking for EPIcode.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
6.7	Test procedures or plans specify the following, as applicable: (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.	No.	See Criterion 6.6 summary remarks.

4.6.2 Sources and Method of Review

Documentation supplied or referenced with the software package and the software developer's partial response to the software information template shown in Appendix A were used as the basis for response to this requirement.

4.6.3 Software Quality-Related Issues or Concerns

Lack of a verifiable, written set of SQA plans and procedures, which includes test reports, for EPIcode should be addressed.

4.6.4 Recommendations

Recommendations related to this topical area are provided as follows:

- It is recommended that benchmark comparisons and validation cases be formally documented (current documentation is in the form of sample case illustrations in the user's manual for the previous version of the code).
- It is recommended that formal test report documentation be established for future upgrades to the code.

4.7 Topical Area 7 Assessment: User Instructions

This area corresponds to the requirement entitled User Instructions in Table 3-3 of (DOE 2003e).

4.7.1 Criterion Specification and Result

Table 4.7-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

Table 4.7-1 — Subset of Criteria for User Instructions Topic and Results

Criterion Number	Criterion Specification	Compliant	Summary Remarks
7.1	A description of the model is documented and made available to users.	Yes.	EPIcode strictly follows the well-established Gaussian model. EPIcode uses no "black-box" techniques. All algorithms are presented and fully referenced in the onboard Software User Documentation.
7.2	User's manual or guide describes software and hardware limitations and identifies includes approved operating systems (for cases where source code is provided, applicable compilers should be noted).	Yes.	(EPIcode, 2003)
7.3	User's manual or guide includes description of the user's interaction with the software.	Yes.	(EPIcode, 2003)
7.4	User's manual or guide includes a description of any required training necessary to use the software.	Not Applicable.	The user's manual does not state the need for any required general training. Formal training, while recommended, is not required.
7.5	User's manual or guide includes input and output specifications.	Yes.	(EPIcode, 2003)
7.6	User's manual or guide includes a description of user messages initiated as a result of improper input and how the user can respond.	Partial.	The user's documentation content is too brief on potential user-induced software problems. Common errors and warning messages could be included with suggested solutions. For some parameters, EPIcode will only allow values within a certain range that is identified in the dialog box that prompts the user to enter input. If the user attempts to input data outside the range, EPIcode will set the value to either the minimum or maximum value of the allowable range as appropriate for the attempted input. It is recommended that a warning message be given when the release

Criterion Number	Criterion Specification	Compliant	Summary Remarks
			scenario has conditions that might lead to dense gas type of atmospheric transport and dispersion.
7.7	User's manual or guide includes information for obtaining user and maintenance support.	Yes.	(EPIcode, 2003)

4.7.2 Sources and Method of Review

Documentation supplied or referenced with the software package and the software developer's partial response to the software information template shown in Appendix A were used as the basis for response to this requirement.

4.7.3 Software Quality-Related Issues or Concerns

User instruction documentation is good. No substantive issues or concerns have surfaced.

4.7.4 Recommendations

Recommendations related to this topical area are as follows:

- The user's documentation content is too brief on potential user-induced software problems. Common errors and warning messages could be included with suggested solutions. Additionally, it is recommended that a warning message be given when the release scenario has conditions that might lead to dense gas type of atmospheric transport and dispersion.

4.8 Topical Area 8 Assessment: Acceptance Test

This area corresponds to the requirement entitled Acceptance Test Table 3-3 of (DOE 2003e). During this phase of the software development, the software becomes part of a system incorporating applicable software components, hardware, and data and is accepted for use. Much of this testing is the burden of the user organization, but the developing organization shoulders some responsibility.

4.8.1 Criterion Specification and Result

Table 4.8-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

Table 4.8-1 — Subset of Criteria for Acceptance Test Topic and Results

Criterion Number	Criterion Specification	Compliant	Summary Remarks
8.1	To the extent applicable to the developer, acceptance testing includes a comprehensive test in the operating environment(s).	Uncertain.	A verifiable, written set of SQA plans and procedures, which would include acceptance testing documentation, is lacking for EPIcode.
8.2	To the extent applicable to the developer acceptance testing was performed prior to approval of the computer program for use.	Uncertain.	See Criterion 8.1 summary remarks.
8.3	The acceptance testing comprehensively evaluates software performance against specified software requirements. To the extent applicable to the developer software validation was performed to ensure that the installed software product satisfies the specified software requirements.	Yes.	EPIcode has an automatic QC check to ensure correct installation and operation of the software. Selection of this option automatically runs all of the EPIcode Release Examples/Case Studies (see onboard Documentation), to verify correct EPIcode operation. Each Example is executed with all parameters/defaults set to the exact values stated in the documentation. The resulting output is compared with the documented results. This ensures that EPIcode has been installed and is operating correctly.
8.4	Acceptance testing documentation includes results of the execution of test cases for system installation and integration, user instructions (Refer to Requirement 7 above), and documentation of the acceptance of the software for operational use.	Yes.	See above.

4.8.2 Sources and Method of Review

Documentation supplied or referenced with the software package and the software developer's partial response to the software information template shown in Appendix A were used as the basis for response to this requirement.

4.8.3 Software Quality-Related Issues or Concerns

Lack of a verifiable, written set of SQA plans and procedures, which include acceptance testing documentation for EPIcode should be addressed.

4.8.4 Recommendations

Recommendations related to this topical area are provided as follows:

- A documented implementation process will be needed for EPIcode to meet all prerequisites for the DOE toolbox.

4.9 Topical Area 9 Assessment: Configuration Control

This area corresponds to the requirement entitled Configuration Control in Table 3-3 of (DOE 2003e).

4.9.1 Criterion Specification and Result

Table 4.9-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

Table 4.9-1 — Subset of Criteria for Configuration Control Topic and Results

Criterion Number	Criterion Specification	Compliant	Summary Remarks
9.1	For the developer, the methods 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) are described in implementing procedures.	Uncertain.	Because a written set of SQA plans and procedures, which would include configuration control procedures, is lacking for EPIcode, a thorough evaluation was not possible.
9.2	Implementing procedures meet applicable criteria for configuration identification, change control and configuration status accounting.	Uncertain.	See Criterion 9.1 summary remarks.

4.9.2 Sources and Method of Review

Documentation supplied or referenced with the software package and the software developer's partial response to the software information template shown in Appendix A were used as the basis for response to this requirement.

4.9.3 Software Quality-Related Issues or Concerns

Lack of a verifiable, written set of SQA plans and procedures, which include configuration control documentation, for EPIcode should be addressed.

4.9.4 Recommendations

Recommendations related to this topical area are provided as follows:

- A documented configuration control process will be needed for EPIcode to meet all prerequisites for the DOE toolbox.

4.10 Topical Area 10 Assessment: Error Impact

This area corresponds to the requirement entitled Error Impact in Table 3-3 of (DOE 2003e).

4.10.1 Criterion Specification and Result

Table 4.10-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

Table 4.10-1 — Subset of Criteria for Error Impact Topic and Results

Criterion Number	Criterion Specification	Compliant	Summary Remarks
10.1	The developing organization's problem reporting and corrective action process addresses the appropriate requirements of its corrective action system and is documented in implementing procedures.	Partial.	Homann Associates, Inc. controls the error notification and corrective actions process. No written confirmation of a documented process.
10.2	The process for evaluating, and documenting whether a reported problem is an error is documented and implemented.	Partial.	No written confirmation of a documented process. Only given an example of the process as it relates to a recent incident and corrective action: Revised EPA Evaporation model in EPIcode. Homann Associates was notified by LLNL NARAC that the EPA Evaporation model had been revised. Homann Associates reviewed/revised the Evaporation model per EPA document "Risk Management Program Guidance for Offsite Consequence Analysis," United States Environmental Protection Agency, EPA 550-B-99-009, April 1999. Appendix D – Technical Background, pg. D-2.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
			The mass transfer coefficient of water is now assumed to be 0.67; The value of 0.67 is based on the Donald MacKay and Ronald S. Matsugu,

Table 4.10-1 — Subset of Criteria for Error Impact Topic and Results (Continued)

Criterion Number	Criterion Specification	Compliant	Summary Remarks
			"Evaporation Rates of Liquid Hydrocarbon Spills on Land and Water," Canadian Journal of Chemical Engineering, August 1973, p. 434. The value of the factor that includes conversion factors, mass coefficient for water, and the molecular weight of water to the one-third power, originally 0.106, is now 0.284. The net result is an evaporation rate that is 2.68 times greater than previous EPIcode versions.
10.3	The process for disposition of the problem reports, including notification to the originator of the results of the evaluation, is documented and implemented.	Uncertain.	Because SQA plans and procedures from the software developer are not available, a thorough evaluation was not possible.
10.4	A documented process provides guidance on determining how identified errors relate to appropriate software engineering elements and is implemented.	Uncertain.	See Criterion 10.3 summary remarks.
10.5	The process is documented and implemented for determining how an error impacts past and present use of the computer program.	Uncertain.	See Criterion 10.3 summary remarks.
10.6	The process is documented and implemented for determining how an error and resulting corrective action impacts previous development activities.	Uncertain.	See Criterion 10.3 summary remarks.
10.7	The process is documented and implemented describing how the users are notified of an identified error, its impact; and how to avoid the error, pending implementation of corrective actions.	Uncertain.	See Criterion 10.3 summary remarks.

4.10.2 Sources and Method of Review

Documentation supplied or referenced with the software package and the software developer's partial response to the software information template shown in Appendix A were used as the basis for response to this requirement.

4.10.3 Software Quality-Related Issues or Concerns

Lack of a verifiable, written set of SQA plans and procedures, which includes error notification and corrective action report, for EPIcode should be addressed.

4.10.4 Recommendations

Recommendations related to this topical area are provided as follows:

- A documented error notification and corrective action process will be needed for EPIcode to meet all prerequisites for the DOE toolbox.

4.11 Training Program Assessment

The software developer's does not have a published training program available for review. It is suggested that training on EPIcode be given at the Energy Facility Contractors Group (EFCOG) conferences. The winter session is during the Safety Basis Subgroup meeting and the summer session is the larger Safety Analysis Working Group, and historically has included training workshops.

4.12 Software Improvements

The EPIcode software was recently upgraded with the issuance of Version 7.0 in September of 2003. EPIcode Version 7.1 is currently in alpha test. This version contains new chemical warfare and biological warfare features. It allows the user to select new output options included time integrated concentration and inhaled dose. A dense gas warning feature is added. A dense gas capability is in development. Additional documentation has been added, inclusive of case studies and validation examples.

It is estimated that a concentrated program to upgrade the SQA pedigree of EPIcode to be compliant with the ten criteria discussed here would require fourteen to sixteen full-time equivalent (FTE)-months. Technical review of the chemical databases associated with this software is assumed to have been performed, and is not included in the level-of-effort estimate.

5.0 Conclusion

The gap analysis for Version 7.0 of the EPIcode software, based on a set of requirements and criteria compliant with NQA-1, has been completed. Of the ten SQA requirements for existing software at the Level B classification (important for safety analysis but whose output is not applied without further review), two requirements are met at acceptable level, i.e., *Classification* (1) and *User Instructions* (7). Improvement actions are recommended for EPIcode to fully meet the remaining eight requirements. This evaluation outcome is deemed acceptable because: (1) EPIcode is used as a tool, and as such its output is applied in safety analysis only after appropriate technical review; (2) User-specified inputs are chosen at a reasonably conservative level of confidence; and (3) Use of EPIcode is limited to those analytic applications for which the software is intended.

Suggested remedial actions for this software would warrant upgrading software documents. The complete list of revised 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 Manual.

Overall, it was determined that the EPIcode 7.0 does meet its intended function for use in supporting documented safety analysis. However, as with all safety-related software, users should be aware of current limitations and capabilities of the software for supporting safety analysis. Informed use of the code can be assisted by appropriate use of current EPIcode documentation and the EPIcode guidance report for DOE safety analysts, *EPIcode Computer Code Application Guidance for Documented Safety Analysis*, (DOE, 2004). Furthermore, while SQA improvement actions are recommended for EPIcode, no evidence has been found of programming, logic, or other types of software errors in EPIcode 7.0 that have led to non-conservatisms in nuclear facility operations, or in the identification of facility controls.

Recommendations are given in Section 2.3 of this document for upgrading the capabilities of EPIcode, focusing on added technical capabilities to broaden the use of EPIcode for DSA-type applications and reducing conservatism in the results.

6.0 Acronyms and Definitions

ACRONYMS:

ASME	American Society of Mechanical Engineers
CD	Compliance Decision
CFR	Code of Federal Regulations
DNFSB	Defense Nuclear Facilities Safety Board
DOE	Department of Energy
DSA	Documented Safety Analysis
EFCOG	Energy Facility Contractors Group
EH	DOE Office of Environment, Safety and Health
EM	DOE Office of Environmental Management
IEEE	Institute of Electrical and Electronics Engineers
IP	Implementation Plan
QAP	Quality Assurance Program (alternatively, Plan)
SQA	Software Quality Assurance
V&V	Verification and Validation
WSRC	Westinghouse Savannah River Company

DEFINITIONS:

The following definitions are taken from the Implementation Plan. References in brackets following definitions indicate the original source, when not the Implementation Plan.

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

Central Registry — An organization designated to be responsible for the storage, control, and long-term maintenance of the Department's safety analysis "toolbox codes." The central registry may also perform this function for other codes if the Department determines that this is appropriate.

Classification (Level of Software) — Determination of the level of software quality assurance associated with a computer code commensurate with the importance of the software application. For the toolbox codes, classification level is determined as described in Appendix A of: "Software Quality Assurance Plan and Criteria for the Safety Analysis Toolbox Codes".

Computer Code — A set of instructions that can be interpreted and acted upon by a programmable digital computer (also referred to as a module or a computer program).

Configuration Item — A collection of hardware or software elements treated as a unit for the purpose of configuration control. [NQA-1]

Configuration Management —The process that controls the activities, and interfaces, among design, construction, procurement, training, licensing, operations, and maintenance to ensure that the configuration of the facility is established, approved and maintained. (Software specific): The process of identifying and defining the configuration items in a system (i.e., software and hardware), controlling the release and change of these items throughout the system's life cycle, and recording and reporting the status of configuration items and change requests. [NQA-1]

Data Library — A data file for use with an executable code that is created and maintained by the controlling organization and is not intended for modification by the user.

Dedication (of Software) — The evaluation of software not developed under utilizing organization existing QA plans and procedures (or not developed under NQA-1 standards). The evaluation determines and asserts the software's compliance with NQA-1 quality standards and its readiness for use in specific applications. (Typically applies to commercially available software.) The utilizing organization reviews the intended software application sufficiently to determine the critical functions that provide evidence of the software's suitability for use. Once the critical functions have been established, methods are defined to verify critical function adequacy and provide verifiable acceptance criteria. Acceptable dedication methods are implemented and required documentation is prepared.

Design Requirements — Description of the methodology, assumptions, functional requirements, and technical requirements for a software system.

Discrepancy — The failure of software to perform according to its documentation.

Error —A condition deviating from an established base line, including deviations from the current approved computer program and its baseline requirements. [NQA-1]

Executable Code — The user form of a computer code. For programs written in a compilable programming language, the compiled and loaded program. For programs written in an interpretable programming language, the source code.

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]

Gap Analysis — Evaluation of the Software Quality Assurance attributes of specific computer software against identified criteria.

Independent Verification and Validation (IV&V) — Verification and validation performed by an organization that is technically, managerially, and financially independent of the development organization.

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]

Object Code — A computer code in its compiled form. This applies only to programs written in a compilable programming language.

Operating Environment — A collection of software, firmware, and hardware elements that provide for the execution of computer programs. [NQA-1]

Safety Analysis and Design Software — Computer software that is not part of a structure, system, or component (SSC) but is used in the safety classification, design, and analysis of nuclear facilities to ensure proper accident analysis of nuclear facilities; proper analysis and design of safety SSCs; and proper identification, maintenance, and operation of safety SSCs.

Safety-Class Structures, Systems, and Components (SC SSCs) — SSCs, including portions of process systems, whose preventive and mitigative function is necessary to limit radioactive hazardous material exposure to the public, as determined from the safety analyses. [10 CFR 830]

Safety-Significant Structures, Systems, and Components (SS SSCs) — SSCs which are not designated as safety-class SSCs, but whose preventive or mitigative function is a major contributor to defense in depth and/or worker safety as determined from safety analyses. [10 CFR 830] As a general rule of thumb, SS SSC designations based on worker safety are limited to those systems, structures, or components whose failure is estimated to result in prompt worker fatalities, serious injuries, or significant radiological or chemical exposure to workers. The term serious injuries, as used in this definition, refers to medical treatment for immediately life-threatening or permanently disabling injuries (e.g., loss of eye, loss of limb). The general rule of thumb cited above is neither an evaluation guideline nor a quantitative criterion. It represents a lower threshold of concern for which an SS SSC designation may be warranted. Estimates of worker consequences for the purpose of SS

SSC designation are not intended to require detailed analytical modeling. Consideration should be based on engineering judgment of possible effects and the potential added value of SS SSC designation. [DOE G 420.1-1]

Safety Software — Includes both safety system software and safety analysis and design software.

Safety Structures, Systems, and Components (SSCs) — The set of safety-class SSCs and safety-significant SSCs for a given facility. [10 CFR 830]

Safety System Software — Computer software and firmware that performs a safety system function as part of a structure, system, or component (SSC) that has been functionally classified as Safety Class (SC) or Safety Significant (SS). 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 function.

Software — Computer programs, operating systems, procedures, and possibly associated documentation and data pertaining to the operation of a computer system. [IEEE Std. 610.12-1990]

Software Design Verification —The process of determining if the product of the software design activity fulfills the software design requirements. [NQA-1]

Software Engineering — The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software; also: the study of these applications. [NQA-1]

Source Code — A computer code in its originally coded form, typically in text file format. For programs written in a compilable programming language, the uncompiled program.

System Software —Software designed to enable the operation and maintenance of a computer system and its associated computer programs. [NQA-1]

Test Case —A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement. [NQA-1]

Test Case Input — Input data for a test case used to verify a modification to a module or a data library.

Test Plan (Procedure) —A document that describes the approach to be followed for testing a system or component. Typical contents identify the items to be tested, tasks to be performed, and responsibilities for the testing activities. [NQA-1]

Testing —An element of verification for the determination of the capability of an item to meet specified requirements by subjecting the item to a set of physical, chemical, environmental, or operating conditions. [NQA-1]

Testing (Software) —The process of

(a) Operating a system (i.e., software and hardware) or system component under specified conditions;

(b) Observing and recording the results; 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. [NQA-1]

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. Toolbox codes meet minimum quality assurance criteria. They may be applied to support 10 CFR 830 DSAs provided the application domain and input parameters are valid. In addition to public domain software, commercial or proprietary software may also be considered. In addition to safety analysis software, design codes may also be included if there is a benefit to maintain centralized control of the codes [modified from DOE N 411.1].

User Manual — A document that presents the information necessary to employ a system or component to obtain desired results. Typically described are system or component capabilities, limitations, options, permitted inputs, expected outputs, possible error messages, and special instructions. Note: A user manual is distinguished from an operator manual when a distinction is made between those who operate a computer system (mounting tapes, etc.) and those who use the system for its intended purpose. Syn: User Guide. [IEEE 610-12]

Validation – 1. The process of testing a computer program and evaluating the results to ensure compliance with specified requirements [ANSI/ANS-10.4-1987].
2.The process of determining the degree to which a model is an accurate representation of the real-world from the perspective of the intended uses of the model [Department of Defense Directive 5000.59, *DoD Modeling and Simulation (M&S) Management*].

Verification – 1. The process of evaluating the products of a software development phase to provide assurance that they meet the requirements defined for them by the previous phase [ANSI/ANS-10.4-1987].
2. The process of determining that a model implementation accurately represents the developer's conceptual description and specifications [Department of Defense Directive 5000.59, *DoD Modeling and Simulation (M&S) Management*].

7.0 References

CFR, Code of Federal Regulations (10 CFR 830). 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, (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 (2002). *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 (2003a). *Implementation Plan for Defense Nuclear Facilities Safety Board Recommendation 2002-1: Quality Assurance for Safety Software at Department of Energy Nuclear Facilities*, Report, (March 13, 2003).

DOE, U.S. Department of Energy (2003b). *Designation of Initial Safety Analysis Toolbox Codes*, Letter, (March 28, 2003).

DOE, U.S. Department of Energy (2003c). *Assessment Criteria and Guidelines for Determining the Adequacy of Software Used in the Safety Analysis and Design of Defense Nuclear Facilities*, Report, CRAD-4.2.4-1, Rev 0, (August 27 2003).

DOE, U.S. Department of Energy (2003d). *Software Quality assurance Improvement Plan: Format and Content For Code Guidance Reports*, Revision A (draft), Report, (August 2003).

DOE, U.S. Department of Energy (2003e). *Software Quality Assurance Plan and Criteria for the Safety Analysis Toolbox Codes*, Revision 1, (November 2003).

DOE, U.S. Department of Energy (2004). EPIcode Computer Code Application Guidance for Documented Safety Analysis, (May 2004).

EPA, U.S. Environmental Protection Agency (1987). *Technical Guidance for Hazards Analysis: Emergency Planning for Extremely Hazardous Substances*, U.S. Environmental Protection Agency, Federal Emergency Management Agency, U.S. Department of Transportation (December 1987).

EPA, U.S. Environmental Protection Agency (1999). *Risk Management Program Guidance for Offsite Consequences*, EPA550-B-99-009, Appendix D – Technical Background (April, 1999).

EPIcode (2003). *EPIcode Version 7.0 User Documentation*, Online Help distributed with software package, Homann Associates, Inc. (September 2003).

S. G. Homann (1996), *EPIcode User's Guide, Version 6.0*, Homann Associates, Inc.

Appendices

Appendix	Subject
A	Software Information Template

APPENDIX A.— SOFTWARE INFORMATION TEMPLATE

Information Form

Development and Maintenance of Designated Safety Analysis Toolbox Codes

The following summary information in Table 2 should be completed to the level that is meaningful – enter N/A if not applicable. *(Note: This information is provided to give the reader of this Gap report, an idea of the information requested to complete the Gap analysis for EPIcode. Detailed information in response was not filled in. See Section 1.6. Instead, the contacts and the Gap authors used the form as a guide for continual discussion throughout the Gap analysis for EPIcode.)*

Table 2. Summary Description of Subject Software

Table 2. Summary Description of Subject Software	
Type	**Specific Information**
Code Name	
Version of the Code	
Developing Organization and Sponsor Information	
Auxiliary Codes	
Software Platform/Portability	
Coding and Computer(s)	
Technical Support Point of Contact	
Code Procurement Point of Contact	
Code Package Label/Title	
Contributing Organization(s)	

Table 2. Summary Description of Subject Software	
Type	**Specific Information**
Recommended Documentation - Supplied with Code Transmittal upon Distribution or Otherwise Available	1. 2. 3. 4. 5.
Input Data/Parameter Requirements	
Summary of Output	
Nature of Problem Addressed by Software	
Significant Strengths of Software	
Known Restrictions or Limitations	
Preprocessing (set-up) time for Typical Safety Analysis Calculation	
Execution Time	
Computer Hardware Requirements	
Computer Software Requirements	
Other Versions Available	

Table 2. Summary Description of Subject Software	
Type	**Specific Information**

Table 3. Point of Contact for Form Completion

Individual(s) completing this information form: Name: Organization: Telephone: Email: Fax:	

1. Software Quality Assurance Plan

The software quality assurance plan for your software may be either a standalone document, or embedded in other documents, related procedures, QA assessment reports, test reports, problem reports, corrective actions, supplier control, and training package.

1.a For this software, identify the governing Software Quality Assurance Plan (SQAP)? [Please submit a PDF of the SQAP, or send hard copy of the SQAP[2]]

1.b What software quality assurance industry standards are met by the SQAP?

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

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

1.e Is the SQAP proceduralized in your organization? If so, please list the primary procedures that provide guidance.

Guidance for SQA Plans:

Requirement 2 – SQA Procedures/Plans (Table 3-2 of SQA Plan/Criteria (DOE, 2003a)
ASME NQA-1 2000 Section 200

[2] Notify Kevin O'Kula of your intent to send hard copies of requested reports and shipping will be arranged.

IEEE Standard 730, *IEEE Standard for Software Quality Assurance Plans*.
IEEE Standard 730.1, *IEEE Guide for Software Quality Assurance Planning*.

2. Software Requirements Description

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

2.a **For this software, was a software requirements description documented with the software sponsor?** [If available, please submit a PDF of the Software Requirements Description, or include hard copy with transmittal of SQAP]

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

Guidance for Software Requirements Documentation:

Requirement 5 – SQA Procedures/Plans (Table 3-2 of SQA Plan/Criteria (DOE, 2003a))
ASME NQA-1 2000 Section 401
IEEE Standard 830, *Software Requirements Specifications*

3. Software Design Documentation

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

The SDD should provide the following:

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

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

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

Guidance for Software Design Documentation:

Requirement 6 – SQA Procedures/Plans (Table 3-2 of SQA Plan/Criteria (DOE, 2003a))
ASME NQA-1 2000 Section 402
IEEE Standard 1016.1, *IEEE Guide for Software Design Descriptions* IEEE Standard 1016-1998, *IEEE Recommended Practice for Software Design Descriptions*
IEEE Standard 1012, *IEEE Standard for Software Verification and Validation;*
IEEE Standard 1012a, *IEEE Standard for Software Verification and Validation – Supplement to 1012*

4. Software User Documentation

Software User Documentation is necessary to assist the user in installing, operating, managing, and maintaining the software, and to ensure that the software satisfies user requirements. At minimum, the documentation should describe:

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

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

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

4.c Training – How is training offered in correctly running the subject software? Complete the appropriate section in the following:

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

Guidance for Software User Documentation:

Requirement 9 – SQA Procedures/Plans (Table 3-2 of SQA Plan/Criteria (DOE, 2003a))
ASME NQA-1 2000 Section 203
IEEE Standard 1063, *IEEE Standard for Software User Documentation*

5. Software Verification &Validation Documentation (Includes Test Reports)

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

(a) The software adequately and correctly performs all intended functions, and
(b) The software does not perform any unintended function.

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

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

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

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

5.c Testing of software: What has been used to test the subject software?

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

Provide any reports or written documentation substantiating the responses above.

Guidance for Software Verification & Validation, and Testing Documentation:

Requirement 6 – *Design Phase* - SQA Procedures/Plans (Table 3-2 of SQA Plan/Criteria (DOE, 2003a))
Requirement 8 – *Testing Phase* - SQA Procedures/Plans (Table 3-2 of SQA Plan/Criteria (DOE, 2003a))

Requirement 10 – *Acceptance Test* - SQA Procedures/Plans (Table 3-2 of SQA Plan/Criteria (DOE, 2003a))
ASME NQA-1 2000 Section 402 (Note: Some aspects of verification may be handled as part of the Design Phase).
ASME NQA-1 2000 Section 404 (Note: Aspects of validation may be handled as part of the Testing Phase).
IEEE Standard 1012, *IEEE Standard for Software Verification and Validation;*
IEEE Standard 1012a, *IEEE Standard for Software Verification and Validation – Supplement to 1012*
IEEE Standard 829, *IEEE Standard for Software Test Documentation*.
IEEE Standard 1008, *Software Unit Testing*

6. Software Configuration Management (SCM)

A process and related documentation for SCM should be defined, maintained, and controlled.

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

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

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

6.a For the subject software, has a Software Configuration Management Plan been prepared, or are its constituent parts covered elsewhere? [If available, please submit a PDF of the Software Configuration Management Plan and related procedures, or include hard copies with transmittal of SQAP].

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

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

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

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

Requirement 12 – *Configuration Control* - SQA Procedures/Plans (Table 3-2 of SQA Plan/Criteria (DOE, 2003a))
ASME NQA-1 2000 Section 203
IEEE Standard 828, *IEEE Standard for Software Configuration Management Plans*.

7. Software Problem Reporting and Corrective Action

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

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

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

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

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

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

Category of Error or Defect	Corrective Action	Planned schedule for correc
Major		
Minor		

7.cIdentify the process and procedures governing communication of errors/defects related to the subject software with users.

Guidance for Error/Defect Reporting and Corrective Action Documentation**:**

Requirement 13 – *Error Impact* - SQA Procedures/Plans (Table 3-2 of SQA Plan/Criteria (DOE, 2003a))
ASME NQA-1 2000 Section 204
IEEE Standard 1063, *IEEE Standard for Software User Documentation*

8. Resource Estimates

If one or more plans, documents, or sets of procedures identified in parts one (1) through seven (7) do not exist, please provide estimates of the resources (full-time equivalent (40-hour) weeks, FTE-weeks) and the duration (months) needed to meet the specific SQA requirement.

Enter estimate in Table 4 only if specific document has not been prepared, or requires revision.

Table 4. Resource and Schedule for SQA Documentation

Plan/Document/Procedure	Resource Estimate (FTE-weeks)	Duration of Activity (months)
1. Software Quality Assurance Plan		
2. Software Requirements Document		
3. Software Design Document		
4. Test Case Description and Report		
5. Software Configuration and Control		
6. Error Notification and Corrective Action Report		
7. User's Instructions (User's Manual)		
8. Other SQA Documentation		

Comments or Questions:

9. Software Upgrades

Describe modifications planned for the subject software.

Technical Modifications

Priority	Description of Change	Resource Estimate (FTE-weeks)
1.		
2.		
3.		
4.		
5.		

User Interface Modifications

Priority	Description of Change	Resource Estimate (FTE-weeks)
1.		
2.		
3.		
4.		
5.		

Software Engineering Improvements

Priority	Description of Change	Resource Estimate (FTE-weeks)
1.		
2.		
3.		
4.		
5.		

Other Planned Modifications

Priority	Description of Change	Resource Estimate (FTE-weeks)
1.		
2.		
3.		
4.		
5.		

Thank you for your input to the SQA upgrade process. Your experience and insights are critical towards successfully resolving the issues identified in DNFSB Recommendation 2002-1.

REFERENCES

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 (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 (2002). *Selection of Computer Codes for DOE Safety Analysis Applications* (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*, Letter (March 13, 2003); Report (February 28, 2003).

DOE, U.S. Department of Energy (2003a). *Software Quality Assurance Plan and Criteria for the Safety Analysis Toolbox Codes,* Interim Report, (September 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).