Department of Energy
Washington, DC 20585

December 3, 2003

The Honorable John T. Conway
Chairman
Defense Nuclear Facilities Safety Board
625 Indiana Avenue, NW
Washington, D.C. 20004-2941

Dear Mr. Chairman:

The Implementation Plan for Software Quality Assurance (SQA) in response to Defense Nuclear Facilities Safety Board (DNFSB) Recommendation 2002-1 requires the Office of Environment, Safety and Health (EH) to perform a gap analysis on the toolbox codes. Commitment 4.2.1.3 requires this analysis to determine the actions needed to bring the codes into compliance with SQA criteria and to estimate the resources needed to upgrade each code based on the gap analysis results.

This commitment is partially completed. Three of the six gap analyses with interim reports are complete and are attached. The completed codes include MACCS2, ALOHA, and EPICODE. Work on the remaining three toolbox codes (MELCOR, GENII, and CFAST) is in progress. Delays in establishing a contract mechanism to obtain the proper expertise have extended the commitment delivery date for the remaining three toolbox code gap analyses and final reports to January 2004. This change in schedule should not affect other SQA Implementation Plan commitments.

Please contact me at (202) 586-6151, or have your staff contact Frank Russo at (301) 903-8008 if you have any questions concerning this commitment.

Sincerely,

Beverly A. Cook
Assistant Secretary
Environment, Safety and Health

Attachments (3)

cc: Mark B. Whitaker, DR-1
Frank B. Russo, EH-3
Chip Lagdon, EH-31

SEPARATION

PAGE

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

Software Quality Assurance Improvement Plan: ALOHA Gap Analysis

Interim Report

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

November 2003

INTENTIONALLY BLANK

FOREWORD

This report documents the outcome of an evaluation of the Software Quality Assurance (SQA) attributes of the chemical source term and atmospheric dispersion computer code, ALOHA, 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

INTENTIONALLY BLANK

CONTENTS

INTENTIONALLY BLANK

TABLES

INTENTIONALLY BLANK

FIGURES

Page

None

Software Quality Assurance Improvement Plan: ALOHA Gap Analysis

EXECUTIVE SUMMARY

The Defense Nuclear Facilities Safety Board 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 ALOHA 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 ALOHA code 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 ALOHA against identified criteria.

The balance of this document provides the outcome of the ALOHA 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). Remedial actions are recommended to meet SQA criteria for the remaining eight requirements.

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.

As part of this effort, the draft NOAA theoretical description memorandum for ALOHA 5.0 (Reynolds, 1992), which is the main source of information for technical information, should be updated for recent upgrades, technically reviewed, and issued as final. Once these actions have been accomplished, ALOHA Version 5.2.3 is qualified for the Central Registry. It is estimated that a concentrated program to upgrade the SQA pedigree of ALOHA 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.

INTENTIONALLY BLANK

1.0 Introduction

This document reports on the results of a gap analysis for Version 5.2.3 of the ALOHA 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, ALOHA Version 5.2.3, is likely to require some degree of quality assurance improvement before meeting current SQA standards. The analysis of this document evaluates ALOHA Version 5.2.3 relative to current software quality assurance criteria. It assesses the margin 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 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 will provide information to DOE, code developers, and code users.

DOE will see the following benefits:

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

Each code developer will be provided:

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

DOE safety analysts and code users will benefit from:

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

1.4 Scope

This analysis is applicable to the ALOHA code, one of the six designated toolbox codes for safety analysis. While ALOHA 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 ALOHA code as part of DOE's implementation plan on SQA improvements.

1.6 Methodology for Gap Analysis

The gap analysis for ALOHA 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 ALOHA code. The ten areas are assessed individually in Section 4.

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 as of the date of this interim report, the written response provided by the ALOHA software developers to the information template has been 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). c. Identify documents that are acceptable from SQA perspective. Note inadequate documents as appropriate.
3. Software Product Technical/ Functional Requirements	a. Review requirements documentation to determine if requirements support intended use in Safety Analysis. Document this determination in gap analysis document. b. Review previously conducted software testing to verify that it sufficiently demonstrated software performance required by the Software Requirements Document. Document this determination in the gap analysis document.
4. Testing	a. Determine whether past software testing for the software being evaluated provides adequate assurance that software product/technical requirements have been met. Obtain documentation of this determination. Document this determination in the gap analysis report. b. (Optional) Recommend test plans/cases/acceptance criteria as needed per the SQAP if testing not performed or incomplete.

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

Phase	Procedure
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 5.2.3 of the ALOHA code (NOAA, 1999a). The current version (as of September 2002) of the Areal Locations of Hazardous Atmospheres (ALOHA) code is version 5.2.3, and was released in 1999. ALOHA is a public domain code that is part of a system of software that is known as the Computer-Aided Management of Emergency Operations (CAMEO) that was developed to plan for and respond to chemical emergencies. It is also widely used throughout the DOE complex for safety analysis applications

Specifically, ALOHA 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 duration, and the physical form of the chemical upon release. The analyst specifies the chemical and then characterizes the initial boundary conditions of the chemical with respect to the environment through the source configuration input. The ALOHA code allows for the source to be defined in one of four ways (i.e., direct source, puddle source, tank source, or pipe source) in order to model various accident scenarios. The source configuration input is used to either specify the chemical source term or to provide ALOHA with the necessary information and data to calculate transient chemical release rates and physical state of the chemical upon release.

The ALOHA code considers two classes of atmospheric transport and dispersion based upon the assumed interaction of the released cloud with the atmospheric wind flow.

- For airborne releases in which the initial chemical cloud density is less than or equal to that of the ambient air, ALOHA treats the released chemical as neutrally buoyant.
- Alternatively, if the density of the initial chemical cloud is greater than that of the ambient air, then the possibility exists for either neutrally buoyant or dense-gas type of atmospheric transport and dispersion.

In addition to the source term and downwind concentration calculations, ALOHA allows for the specification of concentration limits for the purpose of consequence assessment (e.g., assessment of human health risks from contaminant plume exposure). ALOHA refers to these concentration limits as

level-of-concern (LOC) concentrations. Safety analysis work uses the emergency response planning guidelines (ERPGs) and temporary emergency exposure limits (TEELs) for assessing human health effects for both facility workers and the general public (Craig, 2001). While ERPGs and TEELs are not explicitly a part of the ALOHA chemical database[2], ALOHA allows the user to input an ERPG or TEEL value as the LOC concentration.

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

Table 1-2 — Summary Description of ALOHA Software

Type	Specific Information
Code Name	ALOHA (Areal Locations of Hazardous Atmospheres)
Version of the Code	Version 5.2.3
Developing Organization and Sponsor Information	DOC/NOAA/NOS Office of Response and Restoration And EPA Office of Emergency Prevention, Preparedness, and Response
Auxiliary Codes	Codes ALOHA is a standalone program but can be used in conjunction with CAMEO and MARPLOT. For more information, see http://response.restoration.noaa.gov
Software Platform/Portability	Available for Macintosh computers running OS 8, OS 9, or OS X; Available for any personal computer that runs Windows 98, 2000, NT, XP, or ME operating systems.
Coding and Computer(s)	C Code
Technical Support Point of Contact	Robert Jones NOAA/ORR 7600 Sand Point Way, Seattle, WA 98115 206-526-4278 Robert.jones@noaa.gov
Code Procurement Point of Contact	A self-extracting installer can be downloaded from: http://www.epa.gov/ceppo/cameo/aloha.htm Mark W Miller DOC/NOAA/NOS/ORR 7600 Sand Point Way, Seattle, WA 98115 206-526-6272 mark.w.miller@noaa.gov
Code Package Label/Title	aloha.exe – Windows alohains.sit.hqx - Macintosh
Contributing Organization(s)	DOC/NOAA/NOS Office of Response and Restoration and EPA Office of Emergency Prevention Preparedness and Response
Recommended Documentation - Supplied with Code Transmittal upon Distribution or Otherwise Available	1. ALOHA MANUAL is a 1.5 MB PDF file (aloha.pdf) that can be downloaded directly from http://www.epa.gov/ceppo/cameo/aloha.htm

[2] The ALOHA chemical database incorporates two sets of concentration limits that are used in the chemical industry to address worker safety issues: (1) immediately dangerous to life or health (IDLH) and (2) threshold limit value – time weighted average (TLV-TWA).

Type	Specific Information
Input Data/Parameter Requirements	The location, and chemical must be selected from scrolling lists. In some cases, the user must specify the concentration level to be displayed. Wind speed, direction, ground roughness, cloud cover, humidity, air temperature, and inversion height must be selected. The inputs needed to specify the source strength depend upon the scenario chosen; the simplest is the direct source and requires the mass or volume release rate.
Summary of Output	Output is provided in text and graphical form, including - rate at which the pollutant is entering the atmosphere as a function of time - indoor and outdoor concentrations as a function of time at a user-defined location - spatial distribution corresponding to the condition that the maximum concentration exceeds a user-specified level of concern
Nature of Problem Addressed by Software	ALOHA provides conservative estimates of the spatial distribution of the peak concentration of a pollutant following an acute release. To accomplish this, ALOHA contains an extensive database of chemical properties, models for estimating the amount of material entering the atmosphere for a wide range of scenarios, and Gaussian and dense gas (based on DEGADIS) dispersion models.
Significant Strengths of Software	ALOHA contains an extensive database of chemical properties so no additional information beyond the chemical identity is required. ALOHA has submodels for estimating the amount of pollutant entering the atmosphere (source strength). ALOHA has a dispersion model capable of accounting for the gravity effects on dense gas dispersion. ALOHA displays uncertainty associated with wind direction. ALOHA's interface is designed to assist users by including intelligent default entries where appropriate, reasonableness checks for input and context sensitive helps which include data entry guidance.
Known Restrictions or Limitations	ALOHA is designed to estimate the airborne concentration of pollutantsover a relatively short time, one hour, and short spatial extent, 10 kilometers. With this restriction, the use of steady-state meteorology is acceptable. ALOHA does not account for steering by local topography, particulates, or reactions (including fire).
Preprocessing (set-up) time for Typical Safety Analysis Calculation	5 - 15 minutes
Execution Time	1 – 10 seconds
Computer Hardware Requirements	Any computer capable of running the operating systems noted above can run ALOHA.
Computer Software Requirements	None
Other Versions Available	N/A

Type	Specific Information
Individual(s) completing this information form: Name: Organization: Telephone: Email: Fax:	 Mark W Miller DOC/NOAA/NOS/ORR 206-526-6272 mark.w.miller@noaa.gov 206-526-6329

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

Table 1-3 — Software Documentation Reviewed for ALOHA

No.	Information	
1.	Ref:	*ALOHA User's Manual* (NOAA, 1999a)
	Remarks:	
2.	Ref:	*ALOHA 5.2.3 Online Help* (NOAA, 1999b)
	Remarks:	
3.	Ref:	*ALOHA Theoretical Description* (Reynolds, 1992)
	Remarks:	
4.	Ref:	*ALOHA User's and ARCHIE: A Comparison*, Report No. HAZMAT 93-2 (M. Evans, 1993)
	Remarks:	
5.	Ref:	http://www.nwn.noaa.gov/sites/hazmat/cameo/alotech/quality.html
	Remarks:	
6.	Ref:	http://www.nwn.noaa.gov/sites/hazmat/cameo/aloha.html
	Remarks:	
7.	Ref:	http://www.epa.gov/ceppo/cameo/instruct.htm
	Remarks:	
8.	Ref:	http://response.restoration.noaa.gov/cameo/aloha.html
	Remarks:	
9.	Ref:	http://response.restoration.noaa.gov/cameo/alohafaq/history.html
	Remarks:	
10.	Ref:	
	Remarks:	

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 ALOHA SQA program, in general, met criteria for Software Classification and User Instructions, Requirements 1 and 7, respectively. Some topical quality areas were not met satisfactorily and they are listed below in Section 2.2 (Exceptions to Requirements).

2.2 Exceptions to Requirements

Some of the more important exceptions to criteria found for ALOHA 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.

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

No.	Criterion	Reason Not Met	Remedial action(s)
1.	SQA Procedures/Plans	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	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	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. As part of this effort, the draft NOAA theoretical description memorandum for ALOHA 5.0 (Reynolds, 1992), which is the main source of information for technical information, should be updated for recent upgrades, technically reviewed, and issued as final.
4.	Testing Phase	A Software Testing Report Document does not exist for review. The documentation of results from validation and benchmark activities are incomplete and in the form of summaries that are found at ALOHA	A Software Testing Report Document should be prepared and made available for review.

No.	Criterion	Reason Not Met	Remedial action(s)
		websites.	
5.	Configuration Control	A Configuration and Control Document does not exist for review.	A Configuration and Control Document should be prepared and made available for review.
6.	Error Notification	An Error Notification and Corrective Action Report does 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 number of improvements that could be made related to the code and its quality assurance. Some of the important ones are listed in Table 2-2.

Table 2-2 — Summary of Important Recommendations for ALOHA

No.	Recommendation
1.	Correct a reported IDLH bug (e-mail to Mark Miller at NOAA on 11/13/2003). The footprint information gives results for the distance that corresponds to the maximum threat zone for IDHL. When the centerline concentration output is requested at this distance, the concentration results are expected to be the IDLH concentration or very close to it. This is not always the case. (Note: The footprint information output seems to be the source of problem, and neither footprint output or IDLH data are typically not used in DSA applications.)
2.	Provide method to write-protect the Chemical Library. In previous versions of ALOHA, the Chemical Library was protected from inadvertent changes by requiring the use of another program, ChemManager. In the current version, this is not the case; permanent changes may be made within ALOHA code itself. This allows any user to permanently change the chemical library. This is especially problematic, in that users have previously been allowed to make changes knowing that they could not alter the chemical library itself. Allowing some method of protecting the chemical library would be beneficial. Although this can be done within the operating system itself by write protecting the ChemLib file, not all users will be knowledgeable enough to know this, and not all installations will write protect the file.
3.	Add capability to model release durations that are greater than one hour and downwind distances that are greater than 10 km. Although we recognize the purpose of this limitation, for safety analysis purposes, it is standard procedure to model releases using persistent meteorology and a straight-line Gaussian plume to a receptor at the site boundary. As many DOE sites are quite large (hundreds of square miles), this forces an analyst to use another tool to perform the same task. Rather than increasing the limit, we would rather it be removed altogether. While this may allow for unrealistic real-time use, it is typically required for bounding consequence calculations.
4.	Add capability to output consequences for multiple receptors in a single ALOHA

No.	Recommendation
	run. DSA analyses may a set of several receptors (e.g., 30m, 100m, 500m, 1km etc.) for which consequences must be determined for every postulated accident scenario. Having the ability to get this output without having to perform a run for each receptor would save time and money on performance and review, and decrease the size of documents. In tandem with the above request, the ability to output a graph of concentration versus centerline distance would be helpful, especially for elevated releases in which the maximum downwind concentration is desired and the distance where this occurs cannot be known apriori.
5a.	Add capability to directly input vapor pressure rather than the only option being for ALOHA to calculate it from chemical properties. Occasionally, releases must be modeled for chemicals that are not in ALOHA's library. For some chemicals, though not all physical property data needed by ALOHA to calculate the vapor pressure is available, the vapor pressures themselves are available. It would be helpful if a vapor pressure could be directly entered and used by ALOHA to calculate an evaporative source term.
5b.	Add capability so a simpler evaporation model is an option to use (one that did not require quite so much physical property data) when insufficient physical property data is known to use the ALOHA evaporation model. The uncertainty in the release quantity is usually far greater than that in the calculation of evaporative source term so the loss of accuracy would not normally be a problem.
6.	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 50th and 95th percentile results.
7.	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. This will allow more accurate modeling for the majority sites that have surface roughness characteristics that fall in between the two extremes of flat grassland and an urban environment.
8.	Add capability to model dry deposition. A simple point depletion model could serve this purpose.
9.	For puddle modeling, allow model to calculate surface area from input of volume (or mass) and puddle depth. When using the code for planning rather than for response, this would be more useful than the current options of inputting the area or diameter, then the volume, depth, or mass.
10.	Add explosion modeling capability. A number of DOE sites have begun to look at explosive dispersal of toxicological material. It would be useful to be able to use the Gaussian plume model of ALOHA to estimate downwind concentrations.
11.	Reword or remove from the initial screen, the limitation on modeling particulates. As dispersion of small (respirable) particles is similar to that of gases, ALOHA is often used in the DOE complex to model respirable aerosols, including powders. The wording of this limitation, for some customers, unnecessarily calls into question this practice.
12	Update, technically review, and issue as final the draft NOAA theoretical description memorandum for ALOHA 5.0 that is the main source of information for technical information (Reynolds, 1992).
13.	Add capability to use long filenames for ALOHA save files.

2.4 Conclusion Regarding Codes Ability to Meet Intended Function

The ALOHA code 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, ALOHA *Computer Code Application Guidance for Documented Safety Analysis*, (DOE 2003f), 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 ALOHA 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.
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. Cross-Reference of Requirements with Subsection and Entry from DOE (2003e)

Subsection (This Report)	Corresponding Entry Table 3-3 from DOE (2003e) No.	Requirement
4.1	1	Software Classification
4.2	2	SQA Procedures/Plans
4.3	5	Requirements Phase
4.4	6	Design Phase
4.5	7	Implementation Phase
4.6	8	Testing Phase
4.7	9	User Instructions
4.8	10	Acceptance Test
4.9	12	Configuration Control
4.10	13	Error Notification

4.1 Topical Area 1 Assessment: Software Classification

This area corresponds to the requirement entitled Software Classification in Table 3-2 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 ALOHA 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	Yes	It is concluded that sufficient information is provided with the documentation that is transmitted

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	classification of the software.		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, 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 with the software package (plus information on ALOHA websites) was 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 developers, 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 ALOHA, the lack of written documentation prevents an independent evaluator from making a definitive confirmation. Based on discussions with the code developer, organizational management of the ALOHA development probably ensured that many elements of a compliant SQA program were fulfilled in an informal manner.

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	A verifiable, written set of SQA plans and procedures is lacking for ALOHA.
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 that 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 with the software package (plus information on ALOHA websites) and limited discussions with the code developer 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 ALOHA 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 ALOHA 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. It is a widely used computer code, which demonstrates that it serves the needs of many analysts. The code is regularly upgraded to improve capabilities.
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 ALOHA.
3.3	Requirements define the functions to be performed by the software and provide detail and information necessary to design the software.	Yes	Information sources for the technical details of the ALOHA algorithms are given in the ALOHA User's manual (NOAA, 1999a), the online help with ALOHA 5.2.3 (NOAA, 1999b), a NOAA report (Evans, 1993) and a draft NOAA theoretical description memorandum (for ALOHA 5.0) (Reynolds, 1992). Information from ALOHA websites is also available. The ALOHA code uses the well-established models, such as the Gaussian puff and plume models. The draft NOAA theoretical description memorandum (for ALOHA 5.0) comprehensively documents these models (Reynolds, 1992). The document, however, is in draft form and should be updated to reflect upgrades that have been made over the past ten years.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
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.	Yes	The online user's documentation implicitly states requirements. The user's documentation also addresses installation and design inputs.
3.5	Acceptance criteria are established in the software requirements documentation for each of the identified requirements.	No	See Criterion 3.2 summary remarks.

4.3.2 Sources and Method of Review

Documentation supplied with the software package (plus information on ALOHA websites) and limited discussions with the code developer were used as the basis for response to this requirement. The draft NOAA theoretical description memorandum (for ALOHA 5.0) is the main source of information for technical information (Reynolds, 1992).

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 ALOHA should be addressed.

4.3.4 Recommendations

Recommendations related to this topical area are provided as follows:

- Formal documentation of the software requirements as in intended in ALOHA 5.2.3 is not required at this time as these requirements can be largely inferred from existing documentation. Documented software requirements, however, will be needed for ALOHA to meet all prerequisites for the DOE toolbox.
- The draft NOAA theoretical description memorandum (for ALOHA 5.0) is the main source of information for technical information (Reynolds, 1992). It should be updated for recent upgrades, technically reviewed, and issued as final.

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.	Possibly. No written confirmation.	Because SQA plans and procedures from the software developer are not available, a thorough evaluation was not possible.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
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.	Possibly. No written confirmation.	See Criterion 4.1 summary remarks.
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).	Possibly. No written confirmation.	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.	Possibly. No written confirmation.	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 events that can affect the computer program.	Possibly. No written confirmation.	See Criterion 4.1 summary remarks.
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 ALOHA.
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 ALOHA user documentation contains this information.
4.9	A Software Design Document, or equivalent, is available and contains the	No	See Criterion 4.6 summary remarks.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	design described in a manner that can be translated into code.		
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.	Possibly. No written confirmation.	While some elements of this criterion may have been met informally per discussions with the software developer, there is no written documentation that allows confirmation.
4.12	The organization responsible for the design assured that the test results adequately demonstrated that the requirements were met.	Possibly. No written confirmation.	See Criterion 4.1 summary remarks.
4.13	The Independent Review (IR) 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.	Possibly. No written confirmation.	While some elements of this criterion may have been met informally per discussions with the software developer, there is no written documentation that allows confirmation.
4.14	The results of the IR are documented with the identification of the verifier indicated.	Possibly. No written confirmation.	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.	Possibly. No written confirmation	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 IR and the methods chosen are shown to be a function of: ➢ The importance to safety, ➢ The complexity of the software,	Possibly. No written confirmation	See Criterion 4.1 summary remarks.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	➢ The degree of standardization, and ➢ The similarity with previously proven software.		

4.4.2 Sources and Method of Review

Documentation supplied with the software package (plus information on ALOHA websites) and limited discussions with the code developer were used as the basis for response to this requirement. The draft NOAA theoretical description memorandum (for ALOHA 5.0) is the main source of information for technical information (Reynolds, 1992).

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 ALOHA should be addressed.

4.4.4 Recommendations

Recommendations related to this topical area are provided as follows:

- Formal documentation of the software design as in intended in ALOHA 5.2.3 may or may not be required at this time. More information is needed from the software developer in order to make this determination. Documented software design, however, will be needed for ALOHA to meet all prerequisites for the DOE toolbox.
- The draft NOAA theoretical description memorandum (for ALOHA 5.0) is the main source of information for technical information (Reynolds, 1992). It should be updated for recent upgrades, technically reviewed, and issued as final.

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.	Possibly. No written confirmation	Because SQA plans and procedures from the software developer are not available, a thorough evaluation was not possible.
5.2	Implemented software was analyzed to identify and correct errors.	Possibly. No written confirmation	See Criterion 5.1 summary remarks.
5.3	The source code finalized during verification (this phase) was placed under configuration control.	Possibly. No written confirmation	See Criterion 5.1 summary remarks.
5.4	Documentation during verification	No	A verifiable, written set of SQA

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	included a copy of the software, test case description and associated criteria that are traceable to the software requirements and design documentation.		plans and procedures, which would include test case descriptions as well as software requirements and design documentation, is lacking for ALOHA.

4.5.2 Sources and Method of Review

Documentation supplied with the software package (plus information on ALOHA websites) and limited discussions with the code developer 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 ALOHA should be addressed.

4.5.4 Recommendations

Recommendations related to this topical area are provided as follows:

- Formal documentation of the implication process as it relates to ALOHA 5.2.3 may or may not be required at this time. More information is needed from the software developer in order to make this determination. A documented implementation process, however, will be needed for ALOHA 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	Benchmark comparisons have been made with the results from the ARCHIE (FEMA, 1989) and CHEMS-PLUS (Little, 1998) computer models. Results from the benchmark comparisons are not reported (NOAA, 1998). Comparisons with field data were also made with the following results reported (NOAA, 1998). More details on the field data comparisons are given below.
6.2	Testing demonstrated the capability of the software to produce valid results for	Possibly. No written	Because SQA plans and procedures from the software

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	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 and does not perform adverse unintended functions.	confirmation	developer are not available, a thorough evaluation was not possible.
6.3	Testing demonstrated that the compute program properly handles abnormal conditions and events as well as credible failures appropriate warning or error messages are provided to the user when the code is used improperly (e.g., an input is specified outside the acceptable range).	Possibly. No written confirmation	See Criterion 6.2 summary remarks.
6.4	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 ALOHA.
6.5	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 6.4 summary remarks.

Additional Detail

The following provides additional detailed explanation on selected criteria in the above table:

Criterion 6.1 — Details on the comparisons with field data are summarized below (NOAA, 1998).

- Source term prediction for non-boiling pool evaporation – All ALOHA predictions were within 42% of measured evaporation rates.
- Source term prediction for liquefied propane – About 83% of ALOHA predictions were within a factor of two of measured vaporization rates.
- Atmospheric transport and dispersion predictions with Gaussian model – ALOHA predictions of mean downwind concentrations were on average 142% of the measured field data. ALOHA tended to underestimate concentrations at distances of 200 meters or more and overestimate concentrations closer in.
- Atmospheric transport and dispersion predictions with dense-gas model – ALOHA predictions were not compared directly with field measurements, but compared with results from the DEGADIS model that was calibrated to 12 trials from field experiments (Spicer, 1989). ALOHA predictions of mean downwind concentrations were on average 107% of DEGADIS predictions, and about 70% of DEGADIS predictions were within a factor of two of measured field concentrations.
- Atmospheric transport and dispersion predictions with dense-gas model for hydrogen flouride (HF) releases – ALOHA predictions were not compared directly with field measurements, but compared with results from the DEGADIS model that was calibrated to 12 trials from field experiments (Spicer, 1989). ALOHA predictions of mean downwind concentrations were on average 48% of the measured field data.

4.6.2 Sources and Method of Review

Documentation supplied with the software package (plus information on ALOHA websites) and limited discussions with the code developer 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 ALOHA 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 incomplete and in the form of website summary).
- 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.	Partially	The draft NOAA theoretical description memorandum (for ALOHA 5.0) is the main source of information for technical information (Reynolds, 1992). It should be updated for recent

Criterion Number	Criterion Specification	Compliant	Summary Remarks
			upgrades, technically reviewed, and issued as final. Currently, this draft NOAA theoretical description memorandum is not readily available.
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	(NOAA, 1999a; NOAA, 1999b)
7.3	User's manual or guide includes description of the user's interaction with the software.	Yes	(NOAA, 1999a; NOAA, 1999b)
7.4	User's manual or guide includes a description of any required training necessary to use the software.	Not Applicable	Formal training, while recommended, is not required.
7.5	User's manual or guide includes input and output specifications.	Yes	(NOAA, 1999a; NOAA, 1999b)
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.	Yes	(NOAA, 1999a; NOAA, 1999b)
7.7	User's manual or guide includes information for obtaining user and maintenance support.	Yes	(NOAA, 1999a; NOAA, 1999b)

4.7.2 Sources and Method of Review

Documentation supplied with the software package (plus information on ALOHA websites) and limited discussions with the code developer were used as the basis for response to this requirement.

4.7.3 Software Quality-Related Issues or Concerns

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

4.7.4 Recommendations

Recommendations related to this topical area are provided as follows:

- The draft NOAA theoretical description memorandum (for ALOHA 5.0) is the main source of information for technical information on the models (Reynolds, 1992). It should be updated for recent upgrades, technically reviewed, and issued as final.

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).	No	A verifiable, written set of SQA plans and procedures, which would include acceptance testing documentation, is lacking for ALOHA.
8.2	To the extent applicable to the developer acceptance testing was performed prior to approval of the computer program for use.	No	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.	No	See Criterion 8.1 summary remarks.
8.4	Acceptance testing documentation includes results of the execution of test cases for system installation and integration, user instructions (Refer to Requirement 9 above), and documentation of the acceptance of the software for operational use.	No	See Criterion 8.1 summary remarks.

4.8.2 Sources and Method of Review

Documentation supplied with the software package (plus information on ALOHA websites) and limited discussions with the code developer 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 ALOHA should be addressed.

4.8.4 Recommendations

Recommendations related to this topical area are provided as follows:

- Formal documentation of the implication process as it relates to ALOHA 5.2.3 may or may not be required at this time. More information is needed from the software developer in order to make this determination. A documented implementation process, however, will be needed for ALOHA 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 developers, 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.	Possibly. No written confirmation	Because a written set of SQA plans and procedures, which would include configuration control procedures, is lacking for ALOHA, a thorough evaluation was not possible.
9.2	Implementing procedures meet applicable criteria for configuration identification, change control and configuration status accounting.	Possibly. No written confirmation	See Criterion 9.1 summary remarks.

4.9.2 Sources and Method of Review

Documentation supplied with the software package (plus information on ALOHA websites) and limited discussions with the code developer 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 ALOHA should be addressed.

4.9.4 Recommendations

Recommendations related to this topical area are provided as follows:

- Formal documentation of the configuration control process as it relates to ALOHA 5.2.3 may or may not be required at this time. More information is needed from the software developer in order to make this determination. A documented configuration control process, however, will be needed for ALOHA 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	Possibly. No written	NOAA controls the error notification and corrective

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	addresses the appropriate requirements of its corrective action system and is documented in implementing procedures.	confirmation	actions process. A set of SQA plans and procedures from the software developer is lacking, making a thorough evaluation not possible.
10.2	The process for evaluating, and documenting whether a reported problem is an error is documented and implemented.	Possibly. No written confirmation	Upgrades are made to code has errors are discovered, frequently by users. A set of SQA plans and procedures from the software developer is lacking, making a thorough evaluation not possible.
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.	Possibly. No written confirmation	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.	Possibly. No written confirmation	See Criterion 10.4 summary remarks.
10.5	The process is documented and implemented for determining how an error impacts past and present use of the computer program.	Possibly. No written confirmation	See Criterion 10.4 summary remarks.
10.6	The process is documented and implemented for determining how an error and resulting corrective action impacts previous development activities.	Possibly. No written confirmation	See Criterion 10.4 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.	Possibly. No written confirmation	See Criterion 10.4 summary remarks.

4.10.2 Sources and Method of Review

Documentation supplied with the software package (plus information on ALOHA websites) and limited discussions with the code developer 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 ALOHA should be addressed.

4.10.4 Recommendations

Recommendations related to this topical area are provided as follows:

- Formal documentation of the error notification and corrective action process as it relates to ALOHA 5.2.3 may or may not be required at this time. More information is needed from the software developer in order to make this determination. A documented error notification and corrective action process, however, will be needed for ALOHA 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. However, discussions with the software developer indicate that there is an active and frequent training program presented nationally on ALOHA/CAMEO.

Discussions are ongoing for the software developer to provide training 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

Planned improvements to the ALOHA software would involve one or more of the following:

- Error fixes
- Software engineering improvements (speed, user interface, input/output etc.)
- Technical model improvements.

It is estimated that a concentrated program to upgrade the SQA pedigree of ALOHA 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.

The software developers have indicated that an upgrade to ALOHA is planned in the near future. The details of the upgrades will be added to this document once the software developers provide this information.

5.0 Conclusion

The gap analysis for Version 5.2.3 of the ALOHA software, based on a set of requirements and criteria compliant with NQA-1, has been completed. Of the ten SQA requirements for existing software classified as level B (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).

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.

As part of this effort, the draft NOAA theoretical description memorandum for ALOHA 5.0 (Reynolds, 1992), which is the main source of information for technical information, should be updated for recent upgrades, technically reviewed, and issued as final.

Overall, it was determined that the ALOHA code as it currently stands meets its intended function for use in supporting documented safety analysis pending resolution of several software development and documentation issues.

Recommendations are given in Section 2.3 of this document for upgrading the capabilities of ALOHA, focusing on added technical capabilities to:

- broaden the use of ALOHA for DSA-type applications,
- reduce conservatism in the results, and
- make the code easier to use.

6.0 Acronyms and Definitions

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".

Commercial Grade Item — An item satisfying a), b), and c) below:

(a) Not subject to design or specification requirements that are unique to nuclear facilities;
(b) Used in applications other than nuclear facilities;
(c) Ordered from the manufacturer/supplier on the basis of specifications set forth in the manufacturer's published product description (for example, catalog). [IEEE Std. 7-4.3.2-1993]

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]

Control Point —A point in the software life cycle at which specified agreements or control (typically a test or review) are applied to the software configuration items being developed, e.g., an approved baseline or release of a specified document or computer program. [NQA-1]

Commercial Grade Dedication —A process of evaluating (which includes testing) and accepting commercial grade items to obtain adequate confidence of their suitability for safety application. [IEEE Std. 7-4.3.2-1993]

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 Analysis Software Group (SASG) — A group of technical experts formed by the Deputy Secretary in October 2000 in response to Technical Report 25 issued by the Defense Nuclear Facilities Safety Board (DNFSB). This group was responsible for determining the safety analysis and instrument and control (I&C) software needs to be fixed or replaced, establishing plans and cost estimates for remedial work, providing recommendations for permanent storage of the software and coordinating with the Nuclear Regulatory Commission on code assessment as appropriate.

Safety-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.

Sample Input — Input data for a designated sample problem which is maintained by the controlling organization for distribution to users.

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 Development Cycle —The activities that begin with the decision to develop a software product and end when the software is delivered. The software development cycle typically includes the following activities:

(a) Software design requirements;
(b) Software design;
(c) Implementation;
(d) Test; and sometimes
(e) Installation. [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]

Software Life Cycle —The activities that comprise the evolution of software from conception to retirement. The software life cycle typically includes the software development cycle and the activities associated with operation, maintenance, and retirement. [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 meeting minimum qualification standards. These codes are sufficiently verified and validated, and may be said to constitute a "safe harbor" methodology. That is to say, the analysts using these codes do not need to present additional defense as to their qualification, provided that they are sufficiently qualified to use the codes and the input parameters are valid.

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 — Assurance that a model as embodied in a computer code is a correct representation of the process or system for which it is intended. This is usually accomplished by comparing code results to either physical data or a validated code designed to perform the same type of analysis. [IEEE-610.12]: The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. Contrast with: **verification**.

Verification — Assurance that a computer code correctly performs the operations specified in a numerical model or the options specified in the user input. This is usually accomplished by comparing code results to a hand calculation or an analytical solution or approximation. [IEEE-610.12]: (1) The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. Contrast with: **validation**. (2) Formal proof of program correctness.

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. See Appendix A for an example of the input to the table prepared for the MACCS2 code.

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	

Appendices

Appendix	Subject
A	Software Information Template

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*, (draft), Report, (September 2003).

DOE, U.S. Department of Energy (2003f). ALOHA Computer Code Application Guidance for Documented Safety Analysis, (draft), Report, (September 2003).

M. Evans (1993). *ALOHA User's and ARCHIE: A Comparison*, Report No. HAZMAT 93-2, Office of Ocean and Resources Conservation and Assessment of the National Oceanic and Atmospheric Administration (NOAA), Seattle, WA.

FEMA (1989). *Handbook of Chemical Hazard Analysis Procedures*, (ARCHIE Manual), Federal Emergency Management Agency (FEMA), U. S. Department of Transportation (USDOT), and U.S. Environmental Protection Agency (USEPA) (1989), Washington, D.C.

A. D. Little (1988). *CHEMS-PLUS (Enhanced Chemical Evaluation Hazard Evaluation Methodologies) Reference Manual*, Version 1.0, Cambridge, MA.

NOAA (1998). *ALOHA Quality Assurance*, National Oceanic and Atmospheric Administration (NOAA), http://www.nwn.noaa.gov/sites/hazmat/cameo/alotech/quality.html.

NOAA (1999a) and EPA. ALOHA User's Manual, Office of Response and Restoration of the National Oceanic and Atmospheric Administration (NOAA) and Chemical Emergency Preparedness and Prevention Office (CEPPO) of the U.S. Environmental (EPA), Seattle, WA.

NOAA (1999b) and EPA. ALOHA 5.2.3 Online Help, Office of Response and Restoration of the National Oceanic and Atmospheric Administration (NOAA) and Chemical Emergency Preparedness and Prevention Office (CEPPO) of the U.S. Environmental (EPA), Seattle, WA.

R. M. Reynolds (1992). ALOHA Theoretical Description, Draft Technical Memorandum NOS ORCA-65 Hazardous Materials Response and Assessment Division (HMRAD) of the National Oceanic and Atmospheric Administration (NOAA), Seattle, WA.

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[3]]

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

[3] 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.aProvide examples of problem/error notification to users and the process followed to address the deficiency. Attach files as necessary.

7.bProvide 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 corre
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).

SEPARATION

PAGE

DOE-EH-4.2.1.3-Interim-EPIcode

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

Interim Report

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

November 2003

INTENTIONALLY BLANK

FOREWORD

This 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

INTENTIONALLY BLANK

CONTENTS

INTENTIONALLY BLANK

TABLES

INTENTIONALLY BLANK

FIGURES

Page

None

Software Quality Assurance Improvement Plan: EPIcode Gap Analysis

EXECUTIVE SUMMARY

The Defense Nuclear Facilities Safety Board 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 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 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 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). Remedial actions are recommended to meet SQA criteria for the remaining eight requirements.

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.

Once these actions have been accomplished, EPIcode Version 7.0 is qualified for the Central Registry. Approximately 14 to 16 full-time equivalent months are estimated to complete these actions.

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 margin 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 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 will provide information to DOE, code developers, and code users.

DOE will see the following benefits:

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

Each code developer will be provided:

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

DOE safety analysts and code users will benefit from:

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

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. The ten areas are assessed individually in Section 4.

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 as of the date of this interim report, the written response provided by the EPIcode software developer to the information template has been 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). c. Identify documents that are acceptable from SQA perspective. Note inadequate documents as appropriate.
3. Software Product Technical/ Functional Requirements	a. Review requirements documentation to determine if requirements support intended use in Safety Analysis. Document this determination in gap analysis document. b. Review previously conducted software testing to verify that it sufficiently demonstrated software performance required by the Software Requirements Document. Document this determination in the gap analysis document.

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

Phase	**Procedure**
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.

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.

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
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.	Information	
1.	Ref:	*EPIcode Version 7.0 User Documentation* (EPIcode, 2003)
	Remarks:	Online Help distributed with software package
2.	Ref:	*Technical Guidance for Hazards Analysis: Emergency Planning for Extremely Hazardous Substances* (EPA, 1987)
	Remarks:	Source of algorithms and methodologies that are used in EPIcode
3.	Ref:	*Risk Management Program Guidance for Offsite Consequences* (EPA, 1999)
	Remarks:	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.	Ref:	*EPIcode User's Guide, Version 6.0* (Homann, 1996)
	Remarks:	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. Some important topical quality areas were not met satisfactorily for the overall SQA pedigree of EPIcode. They are discussed below in Section 2.2 (Exceptions to Requirements).

2.2 Exceptions to Requirements

Some of the more important exceptions to criteria found for EPIcode 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.

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

No.	Criterion	Reason Not Met	Remedial action(s)
1.	SQA Procedures/Plans	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	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	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.	Testing Phase	A Software Testing Report Document does not exist for review.	A Software Testing Report Document should be prepared and made available for review.
5.	Configuration Control	A Configuration and Control Document does not exist for review.	A Configuration and Control Document should be prepared and made available for review.
6.	Error Notification	An Error Notification and Corrective Action Report does 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 number of improvements that could be made related to the code and its quality assurance. Some of the important ones are listed in Table 2-2.

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 50th and 95th 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 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 2003f), 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.
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. Cross-Reference of Requirements with Subsection and Entry from DOE (2003e)

Subsection (This Report)	Corresponding Entry Table 3-3 from DOE (2003e) No.	Requirement
4.1	1	Software Classification
4.2	2	SQA Procedures/Plans
4.3	5	Requirements Phase
4.4	6	Design Phase
4.5	7	Implementation Phase
4.6	8	Testing Phase
4.7	9	User Instructions
4.8	10	Acceptance Test
4.9	12	Configuration Control
4.10	13	Error Notification

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
			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, 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, 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 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	Yes	As stated above, the online user's documentation implicitly states requirements. The user's

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	inputs, design constraints, installation considerations, operating systems (if applicable), and external interfaces necessary to design the software.		documentation also addresses installation and design inputs.
3.5	Acceptance criteria are established in the software requirements documentation for each of the identified requirements.	Partially	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:

- Formal documentation of the software requirements as in intended in EPIcode 7.0 is not required at this time as these requirements can be largely inferred from existing documentation. Documented software requirements, however, 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.	Possibly. No written confirmation.	Because SQA plans and procedures from the software developer are not available, a

Criterion Number	Criterion Specification	Compliant	Summary Remarks
			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.	Possibly. No written confirmation.	See Criterion 4.1 summary remarks.
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).	Possibly. No written confirmation.	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.	Possibly. No written confirmation.	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 events that can affect the computer program.	Possibly. No written confirmation.	See Criterion 4.1 summary remarks.
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	Yes	The EPIcode user documentation contains this information.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	outputs.		
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.	Possibly. No written confirmation.	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 requirements were met.	Possibly. No written confirmation.	See Criterion 4.1 summary remarks.
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.	Possibly. No written confirmation.	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.	Possibly. No written confirmation.	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.	Possibly. No written confirmation	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	Possibly. No written	See Criterion 4.1 summary remarks.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	be a function of: ➢ The importance to safety, ➢ The complexity of the software, ➢ The degree of standardization, and ➢ The similarity with previously proven software.	confirmation	

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:

- Formal documentation of the software design as in intended in EPIcode 7.0 may or may not be required at this time. More information is needed from the software developer in order to make this determination. Documented software design, however, will be needed for EPIcod. .o 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.	Possibly. No written confirmation	Because SQA plans and procedures from the software developer are not available, a thorough evaluation was not possible.
5.2	Implemented software was analyzed to identify and correct errors.	Possibly. No written confirmation	See Criterion 5.1 summary remarks.
5.3	The source code finalized during verification (this phase) was placed under configuration control.	Possibly. No written confirmation	See Criterion 5.1 summary remarks.
5.4	Documentation during verification	No	A verifiable, written set of SQA

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	included a copy of the software, test case description and associated criteria that are traceable to the software requirements and design documentation.		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:

- Formal documentation of the implication process as it relates to EPIcode 7.0 may or may not be required at this time. More information is needed from the software developer in order to make this determination. A documented implementation process, however, 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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
			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.
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 and does not perform adverse unintended functions.	Partially. Not able to confirm all aspects of this requirement	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.
6.3	Testing demonstrated that the computer program properly handles abnormal conditions and events as well as credible failures appropriate warning or error messages are provided to the user when the code is used improperly (e.g., an input is specified outside the acceptable range).	Possibly. No written confirmation	Because SQA plans and procedures from the software developer are not available, a thorough evaluation was not possible.
6.4	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.
6.5	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	No	See Criterion 6.4 summary remarks.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	formatting, and conventions, (9) identification of operating environment, support software, software tools or system software, hardware operating system(s) and/or limitations.		

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	Yes	(EPIcode, 2003)

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	be noted).		
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	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.	No	The EPIcode documentation does not address error messages satisfactorily. 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.
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 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).	No	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.	No	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 as 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:

- Formal documentation of the implication process as it relates to EPIcode 7.0 may or may not be required at this time. More information is needed from the software developer in order to make this

determination. A documented implementation process, however, 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.	Possibly. No written confirmation	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.	Possibly. No written confirmation	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:
Formal documentation of the configuration control process as it relates to EPIcode may or may not be required at this time. More information is needed from the software developer in order to make this determination. A documented configuration control process, however, 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.	Possibly. No written confirmation	Homann Associates, Inc. controls the error notification and corrective actions process.
10.2	The process for evaluating, and documenting whether a reported problem is an error is documented and implemented.	Possibly. No written confirmation of a documented process. Only given an example of the process as it relates to a recent incident, which is summarized in the next column.	**Example that was provided by the code developer of 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. 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, "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.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
			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.	Possibly. No written confirmation	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.	Possibly. No written confirmation	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.	Possibly. No written confirmation	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.	Possibly. No written confirmation	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.	Possibly. No written confirmation	See Criterion 10.4 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:
Formal documentation of the error notification and corrective action process as it relates to EPIcode 7.0 may or may not be required at this time. More information is needed from the software developer in order to make this determination. A documented error notification and corrective action process, however, 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

There are no known planned improvements for the software. The EPIcode software was recently upgraded with the issuance of Version 7.0 in September of 2003.

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 classified as level B (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).

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 software as it currently stands meets its intended function for use in supporting documented safety analysis pending resolution of several software development and documentation issues.

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 reduce conservatism in the results.

6.0 Acronyms and Definitions

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".

Commercial Grade Item — An item satisfying a), b), and c) below:

(a) Not subject to design or specification requirements that are unique to nuclear facilities;
(b) Used in applications other than nuclear facilities;
(c) Ordered from the manufacturer/supplier on the basis of specifications set forth in the manufacturer's published product description (for example, catalog). [IEEE Std. 7-4.3.2-1993]

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]

Control Point —A point in the software life cycle at which specified agreements or control (typically a test or review) are applied to the software configuration items being developed, e.g., an approved baseline or release of a specified document or computer program. [NQA-1]

Commercial Grade Dedication —A process of evaluating (which includes testing) and accepting commercial grade items to obtain adequate confidence of their suitability for safety application. [IEEE Std. 7-4.3.2-1993]

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 Analysis Software Group (SASG) — A group of technical experts formed by the Deputy Secretary in October 2000 in response to Technical Report 25 issued by the Defense Nuclear Facilities Safety Board (DNFSB). This group was responsible for determining the safety analysis and instrument and control (I&C) software needs to be fixed or replaced, establishing plans and cost estimates for remedial work, providing recommendations for permanent storage of the software and coordinating with the Nuclear Regulatory Commission on code assessment as appropriate.

Safety-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.

Sample Input — Input data for a designated sample problem, which is maintained by the controlling organization for distribution to users.

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 Development Cycle —The activities that begin with the decision to develop a software product and end when the software is delivered. The software development cycle typically includes the following activities:

(a) Software design requirements;
(b) Software design;
(c) Implementation;
(d) Test; and sometimes
(e) Installation. [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]

Software Life Cycle —The activities that comprise the evolution of software from conception to retirement. The software life cycle typically includes the software development cycle and the activities associated with operation, maintenance, and retirement. [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 meeting minimum qualification standards. These codes are sufficiently verified and validated, and may be said to constitute a "safe harbor" methodology. That is to say, the analysts using these codes do not need to present additional defense as to their qualification, provided that they are sufficiently qualified to use the codes and the input parameters are valid.

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 — Assurance that a model as embodied in a computer code is a correct representation of the process or system for which it is intended. This is usually accomplished by comparing code results to either physical data or a validated code designed to perform the same type of analysis. [IEEE-610.12]: The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. Contrast with: **verification**.

Verification — Assurance that a computer code correctly performs the operations specified in a numerical model or the options specified in the user input. This is usually accomplished by comparing code results to a hand calculation or an analytical solution or approximation. [IEEE-610.12]: (1) The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. Contrast with: **validation**. (2) Formal proof of program correctness.

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*, (draft), Report, (September 2003).

DOE, U.S. Department of Energy (2003f). EPIcode Computer Code Application Guidance for Documented Safety Analysis, (draft), Report, (September 2003).

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. See Appendix A for an example of the input to the table prepared for the MACCS2 code.

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 3. Point of Contact for Form Completion

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

1. <u>Software Quality Assurance Plan</u>

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 corre
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).

DOE-EH-4.2.1.3-Interim-MACCS2

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

Software Quality Assurance Improvement Plan: MACCS2 Gap Analysis

Interim Report

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

November 2003

INTENTIONALLY BLANK

FOREWORD

This report documents the outcome of an evaluation of the Software Quality Assurance (SQA) attributes of the radiological dispersion computer code, MACCS2, 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

INTENTIONALLY BLANK

CONTENTS

INTENTIONALLY BLANK

TABLES

INTENTIONALLY BLANK

Software Quality Assurance Improvement Plan: MACCS2 Gap Analysis

EXECUTIVE SUMMARY

The Defense Nuclear Facilities Safety Board 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 MACCS2 software, for radiological dispersion and consequence analysis, is one of the codes designated for the toolbox. To determine the actions needed to bring the MACCS2 code 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 an lysis document. The gap analysis evaluates the software quality assurance attributes of MACCS2 against identified criteria.

The balance of this document provides the outcome of the MACCS2 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). Remedial actions are recommended to meet SQA criteria for the remaining eight requirements.

A new software baseline is recommended for MACCS2. Suggested remedial actions for this software would warrant upgrading software documents that describe the new baseline. At minimum, it is recommended that software improvement actions be taken, especially:

1. correcting known defects
2. upgrading user technical support activities
3. providing training on a regular basis, and
4. developing new software documentation.

The complete list of suggested, 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.

Additionally, the User's instruction documentation should be augmented to include error diagnostic advice and suggested inputs for prototypic problem types.

Once these actions have been accomplished, MACCS2 version 1.12 is qualified for the Central Registry. Approximately two full-time equivalent years is estimated to complete these actions.

INTENTIONALLY BLANK

1.0 Introduction

This document reports the results of a gap analysis for Version 1.12 of the MACCS2 computer code. The intent of the gap analysis is to determine the actions needed to bring the specific 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, MACCS2 Version 1.12, will likely require some degree of quality assurance improvement before meeting current SQA standards. The analysis documented herein is an evaluation of MACCS2 relative to current software quality assurance criteria. It assesses the margin 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 bases, by which to evaluate each designated toolbox code. This gap analysis evaluation, 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 input 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 will provide information to DOE, code developers, and code users.

DOE will see the following benefits:

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

Each code developer will be provided:

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

DOE safety analysts and code users will benefit from:

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

1.4 Scope

This analysis is applicable to the MACCS2 code, one of the six designated toolbox codes for safety analysis. While MACCS2 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 in this document 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 MACCS2 code as part of DOE's implementation plan on SQA improvements.

1.6 Methodology for Gap Analysis

The gap analysis for MACCS2 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 utilizes ten of the fourteen topical areas listed in DOE (2003e) related to software quality assurance to assess the quality of the MACCS2 code. The ten areas are assessed individually in Section 4.

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. The core section of the template is attached as Appendix A to the present report. It is noted that as of the date of this interim report, no written response to the information template has been provided by the MACCS2 software developers.

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

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

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

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

1.7 Summary Description of Software Being Reviewed

The gap analysis was performed on Version 1.12 of the MACCS2 code. MACCS2 (Chanin, 1998) is a radiological atmospheric dispersion and consequence code, and is written in FORTRAN 77 and 90. This software is maintained by Sandia National Laboratories (SNL) as an update to MACCS.[2] Since the issuance of DOE-STD-3009-94 for nuclear facility accident analysis, MACCS2 has been used for DOE applications primarily as a tool for deterministic consequence analysis. The output of MACCS2 is used to support decision-making on control selection in nuclear facilities, specifically identification of safety structures, systems, and components (SSCs).

MACCS2 predicts dispersion of radionuclides by the use of multiple, straight-line Gaussian plumes. The direction, duration, sensible heat, and initial radionuclide concentration may be varied from plume to plume. Crosswind dispersion is treated by a multi-step function and both wet and dry depositions features can be modeled as independent processes. For DSA applications, the MACCS2 user can apply either the Latin Hypercube Sampling (LHS) mode or the stratified random sampling mode to process one year of site-specific meteorological data. Based on the meteorological sampling of site-specific data, and application of user-specified dose and/or health effects models, complementary cumulative distribution functions (CCDFs) are calculated for various measures of consequence. The average, median, 95th, and 99.5th percentile doses are provided in the output.

A brief summary of MACCS2 is contained in Table 1-2.

The set of documents reviewed as part of the gap analysis are listed in Table 1-3. The SNL software developers provided references 11 (Proposal to Resolve QA Deficiencies in MACCS2) and 13 (NP 19-1) to support this assessment. Other documentation was previously received from SNL or RSICC.

[2] The United States Nuclear Regulatory Commission (NRC) sponsored the development of the MACCS code (Chanin, 1990; Jow, 1990; Rollstin, 1990; and Chanin, 1993) as a successor to the CRAC2 code for the performance of commercial nuclear industry probabilistic safety assessments (PSAs). The MACCS code was used in the NUREG-1150 PSA study (NRC, 1990a) in the early 1990's. Prior to the code being released to the public, the MACCS code was independently verified by Idaho National Engineering and Environmental Laboratory (Dobbe, 1990). After verification, the NRC released MACCS, Version 1.5.11 for use by the public. Examples of MACCS applied in this period include commercial reactor PSAs (both U.S. and international), as well as non-reactor nuclear facilities (primarily U.S.).

Table 1-2. Summary Description of MACCS2 Software

Type	Specific Information
Code Name	MACCS2 - MELCOR Accident Consequence Code System for the Calculation of the Health and Economic Consequences of Accidental Atmospheric Radiological Releases
Developing Organization and Sponsor	Sandia National Laboratories (SNL) for the U.S. Nuclear Regulatory Commission (primary) and U.S. Department of Energy (minor)
Version of the Code	Version 1.12
Auxiliary Codes	AUXILIARY CODES: DOSFAC2: NRC dose conversion factor (DCF) preprocessor. FGRDCF: DCF preprocessor based on the DCF databases of Federal Guidance Reports 11 and 12 from ORNL (DLC-172). IDCF2: DCF preprocessor based on the IDCF code developed at the Idaho National Engineering Laboratory. COMIDA2: Food pathway preprocessor based on the COMIDA (PSR-343) food pathway preprocessor developed at the Idaho National Engineering Laboratory.
Software Platform/Portability	FORTRAN 77/90, PC based some system dependencies
Coding and Computer	Fortran 77, PC based 80486 or Pentium processor (C00652/PC486/00).
Technical Support	Nathan Bixler Sandia National Laboratories P.O. Box 5800 Albuquerque, NM 87185-0748 (505) 845-3144 nbixler@sandia.gov;
Code Procurement	Radiation Safety Information Computational Center (RSICC) Oak Ridge National Laboratory Post Office Box 2008 Bethel Valley Road Oak Ridge, Tennessee 37831-6171 Phone: 865-574-6176; Fax: 865-241-4046 Email: pdc@ornl.gov
Code Package Identification at RSICC	CCC-652; Included are the references cited below and the Fortran source code, executables and data, which are distributed on 1 CD in self-extracting compressed DOS files.
Contributors	Sandia National Laboratories, Albuquerque, New Mexico, Oak Ridge National Laboratory, Oak Ridge, Tennessee, Idaho National Engineering and Environmental Laboratory, Idaho Falls, Idaho.

Table 1-2. Summary Description of MACCS2 Software (Continued)

Documentation Supplied with Code Transmittal	1. D. Chanin and M. L. Young, "Code Manual for MACCS2, User's Guide," NUREG/CR-6613, Vol. 1, SAND97-0594 (May 1998), Sandia National Laboratories, Albuquerque, NM. 2. D. Chanin and M. L. Young, "Code Manual for MACCS2, Preprocessor Codes COMIDA2, FGRDCF, IDCF2," NUREG/CR-6613, Vol. 2, SAND97-0594 (May 1998), Sandia National Laboratories, Albuquerque, NM.. 3. M. L. Young and D. Chanin, "DOSFAC2 User's Guide," NUREG/CR-6547, SAND97-2776 (December 1997). 4. H-N. Jow, J. L. Sprung, J. A. Rollstin, L. T. Ritchie, D. I. Chanin, "MELCOR Accident Consequence Code System (MACCS), Model Description," NUREG/CR-4691, SAND86-1562, Vol. 2 (February 1990). 5. J. Gregory, "Software Defect Notifications" (May 1998). M. L. Young, "READMAC2.txt" (April 1997). 6. Supplemental: M. L. Young and D. I. Chanin, "DOSFAC2 User's Guide," NUREG/CR-6547 (SAND97-2776, Sandia National Laboratories, Albuquerque, NM.
Nature of Problem	MACCS2 simulates the impact of accidental atmospheric releases of radiological materials on the surrounding environment. This package is a major enhancement of the previous CCC-546/MACCS 1.5.11 package. The principal phenomena considered in MACCS are atmospheric transport, mitigative actions based on dose projection, dose accumulation by a number of pathways including food and water ingestion, early and latent health effects, and economic costs. MACCS can be used for a variety of applications including probabilistic risk assessment (PRA) of nuclear power plants and other nuclear facilities, sensitivity studies to gain a better understanding of the parameters important to PRA, and cost benefit analysis.
Method of Solution	MACCS contains simple models with convenient analytical solutions. A MACCS calculation consists of three phases: input processing and validation, phenomenological modeling and output processing. The phenomenological models are based mostly on empirical data, and the solutions they entail are usually analytical in nature and computationally straightforward. The modeling phase is subdivided into three modules. ATMOS treats atmospheric transport and dispersion of material and its deposition from the air utilizing a Gaussian plume model with Pasquill-Gifford dispersion parameters. EARLY models consequences of the accident to the surrounding area during an emergency action period. CHRONC considers the long term impact in the period subsequent to the emergency action period. Detailed meteorological, population, and economic and health data are required depending upon the type of analyses to be performed and output required. Model parameters can be provided by the user via input facilitating the analysis of consequence uncertainties due to uncertainties in the model parameters.

Table 1-2. **Summary Description of MACCS2 Software (Continued)**

Restrictions or Limitations	The atmospheric model included in the code does not model the impact of terrain effects on atmospheric dispersion. The code also does not model dispersion close to the source (less than 100 meters from the source) or long range dispersion. The economic model included in the code models only the economic cost of mitigative actions.
Run Time	One source term for one meteorological sequence requires approximately 20 seconds on a Pentium 133 MHZ. Running one source term and sampling a year of weather data requires approximately 20 minutes.
Computer Hardware Requirements	IBM compatible 80486 or Pentium PC with 8 MB of RAM. Approximately 30 MB of hard disk space is required to load the complete MACCS2 package. Approximately 11 MB of hard disk space is required to load MACCS2 without the preprocessors included in the MACCS2 package.
Computer Software Requirements	The MACCS2 software was developed in a DOS environment. Lahey F77L-EM/32 Version 5.2 compiler was used to create the executables included in the package, which run successfully in the DOS window of Windows 3.1, Windows95 and WindowsNT. The programs can also be compiled with the Microsoft Powerstation Fortran 1.0a compiler.
Other Versions Available	MACCS 1.5.11.1 (PC486); MACCS 1.5.11.0 (IBM RISC)

Table 1-3 — Software Documentation Reviewed for MACCS2

No.	Reference
1.	Chanin, 1997, D. Chanin and M. Young, *Code Manual for MACCS2: Volume 1, User's Guide; Volume 1-*, NUREG/CR-6613, SAND97-0594, March 1997, Sandia National Laboratories, Albuquerque, NM.
2.	Chanin, 1998, D. Chanin and M. Young, *Code Manual for MACCS2: Volume 1, User's Guide; Volume 2, Pre-Processor Codes COMIDA2, FGRDCF, IDCF2*; May 1998, M. Young, D. Chanin, and V. Banjac, *DOSFAC2 User's Guide*, NUREG/CR-6613, SAND97-0594, May 1998, Sandia National Laboratories, Albuquerque, NM.
3.	Chanin, 1990, D.I. Chanin, J.L. Sprung, L.T. Ritchie, H-N Jow, and J.A. Rollstin, *MELCOR Accident Consequence Code System (MACCS). Volume 1: User's Guide*; H-N Jow, J.L. Sprung, J.A. Rollstin, L.T. Ritchie, and D.I. Chanin, *Volume 2: Model Description*; J.A. Rollstin, D.I. Chanin, and H-N Jow, *Volume 3: Programmer's Reference Manual*; NUREG/CR-4691, Sandia National Laboratories, published by the U.S. Nuclear Regulatory Commission, Washington, DC, 1990.
4.	Chanin, 1992a, D. Chanin, J. Rollstin, J. Foster, and L. Miller, *MACCS Version 1.5.11.1: A Maintenance Release of the Code*, Sandia National Laboratories, Albuquerque, NM, July 14, 1992.
5.	Chanin, 1992b, D.I. Chanin, *A New Emergency Response Model for MACCS*, LA-SUB-94-67, prepared by Teledyne Engineering Consultants, Inc., Albuquerque, NM for Los Alamos National Laboratory, Los Alamos, NM, November 11, 1992.
6.	Dobbe 1990, C.A. Dobbe, E.R. Carlson, N.H. Marshall, E.S. Marwil, J.E. Tolli. *Quality Assurance and Verification of the MACCS Code, Version 1.5,* Idaho National Engineering Laboratory, Idaho Falls, ID, NUREG/CR-5376 (EGG-2566)
7.	DNFSB, 2000, Defense Nuclear Facilities Safety Board, *Quality Assurance for Safety-Related Software at Department of Energy Defense Nuclear Facilities*, Technical Report DNFSB/TECH-25, (January 2000).
8.	WSRC, 1998. Westinghouse Savannah River Company, *MACCS Input Guidance for SRS Applications (U)*, WSRC-RP-98-00978, (October 1998).
9.	East, 1998, J.M. East and E.P. Hope, *Independent Evaluation of the MACCS2 Software Quality Assurance Program (U)*, WSRC-RP-98-00712, Westinghouse Savannah River Company, Aiken, SC (August 1998).
10.	LANL, Los Alamos National Laboratory, *LANL Guidelines for Performing Atmospheric Dispersion Analysis*, Operational Support Tool 300-00-06H, Los Alamos, NM.

Table 1-3 — Software Documentation Reviewed for MACCS2 (continued)

No.	Reference
11.	Bixler, N. 2000, N. Bixler, *Proposal to Resolve QA Deficiencies in MACCS2*, Memorandum to D. Chung (DOE/DP), Sandia National Laboratories, Albuquerque, NM (2000).
12.	DOE 2003f, U.S. Department of Energy. *MACCS2 Computer Code Application Guidance for Documented Safety Analysis*, Interim Report, (September 2003).
13.	SNL 2003, Sandia National Laboratories. Nuclear Waste Management Procedure, NP 19-1, *Software Requirements*, Revision 10, Waste Isolation Pilot Plant, (May 2003).
14.	Summa, F.J., (1996) and F.E. Haskin. *Pre-Release Verification Testing of the MACCS2 Code*. University of New Mexico, Albuquerque, NM
13.	Chanin, D., (1997). Software Quality Assurance Procedures Followed with MACCS2, Letter to K. O'Kula (September 1997).
14.	Gregory, J. (1998). Software Defect Notification. Sandia National Laboratories, Albuquerque, NM (1998).

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 MACCS2 SQA program, in general, met criteria for *Software Classification* and *User Instructions*, Requirements 1 and 7, respectively. Eight topical quality areas were not met satisfactorily. The major deficiency areas are covered below in Section 2.2 (Exceptions to Requirements). Detail on the evaluation process relative to the requirements and the criteria applied are found in Section 4.

2.2 Exceptions to Requirements

Some of the more important exceptions to criteria found for MACCS2 are listed below in Table 2-1. The requirement is given, the reason the requirement was not met is provided, and remedial action(s) are listed to correct the exceptions.

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 Plan and Procedures for Version 1.12 of MACCS2 software were not available for the gap analysis.	As part of the new software baseline, the SQA Plan covering version 1.12 and successor versions of MACCS2 should be provided to the Central Registry and to RSICC. Any SQA procedures that provide prescriptive guidance to the MACCS2 software developers should be made available to a SQA evaluator for confirmatory review.
			• Establish a written and approved SQA plan eliminating draft or non-compliant informal process of development.
			• Upgrade SQA program documentation, especially those procedures used for new features added in MACCS2.
2.	Requirements Phase (Section 4.3)	The Software Requirements Document for Version 1.12 of MACCS2 software has not been finalized.	As part of the new software baseline for MACCS2, a Software Requirements Document should be prepared.
3.	Design Phase (Section 4.4)	A Software Design Document was not made available for the gap analysis. Thus, design information was not directly available. Instead, it was necessary to infer the intent of MACCS2 design from model	As part of the new software baseline for MACCS2, a Software Design Document should be formally prepared.

No.	Criterion	Reason Not Met	Remedial Action(s)
		description and user guidance documents.	
4.	Implementation Phase (Section 4.5)	Written documentation on implementation of Version 1.12 of MACCS2 is not available.	No action needed at this time. The gap analysis inferred from other documentation that source code and other software elements were finalized prior to transmittal of the code to RSICC.
5.	Testing Phase (Section 4.6)	A Software Testing Report Document has not been produced for MACCS2, and therefore, test process and methodology could not be evaluated directly. Thus, testing process and methods had to be inferred from other information. A draft validation study has been previously reported.	As part of the new software baseline for MACSS2, a test case report should be prepared. Some part of the new baseline set of documentation should address the reasonableness of the model for specific source term types, e.g. fire related plumes, deflagration releases, etc.
6.	Acceptance Test (Section 4.8)	An Acceptance Test protocol was not provided to the gap analysis. There is no known formal procedure to assure that an installed version of MACCS2 is working properly.	As part of the new software baseline for MACCS2, an acceptance test process should be documented. This instruction can be made part of an upgraded User's Guide.
7.	Configuration Control (Section 4.9)	A MACCS2 Configuration and Control document was not provided for the gap analysis input, despite indication that this document.	It is recommended that a full-scope Software Configuration and Control document be issued as part of the new software baseline. If this document has been generated, then it should be made available for review.
8.	Error Notification (Section 4.10)	An Error Notification and Corrective Action Report process is in place at SNL, but limited documentation was forwarded to allow a gap analysis to be performed.	While a Software Problem Reporting system was apparently in place at SNL, written documentation should be provided to the Central Registry for verification of its effectiveness.

2.3 Areas Needing Improvement

The gap analysis, communications with DOE, oversight organizations, safety analysts, and inputs from the long-term MACCS/MACCS2 users have identified a number of improvements that could be made related to the code and its quality assurance. The major areas to be addressed are described in this section.

Multiple-plume release. The software upgrade that should be addressed as soon as possible is that impacting calculations containing multiple plume segments (Gregory 1998). Other identified errors in the MACCS2 software, while deserving corrective action as part of good SQA processes and practices, are insignificant relative to most DOE DSA applications.

Multiple versions of MACCS2. There are instances reported of multiple versions of MACCS2 having been disseminated over the last five years. This is not good practice from a software configuration control perspective. It is recommended that all capabilities be made available through one common

distribution site, such as the DOE Central Registry, or the Radiation Safety Information Computational Center (RSICC).

User Interface. Other modifications are recommended on a less urgent basis. Included are improvements to the user interface. MACCS2 still uses a DOS-based operating system, and requires experienced user insights to correctly build an input file. A U.S. NRC-sponsored program will improve this feature by developing a WINDOWS-based system (Bixler, 2000). However, it is unclear whether this modification will be carried over to the mainstream MACCS2 version.

DSA Dispersion/Dose Analysis. Using MACCS2 to quantify 95th percentile direction-independent doses to receptors at non-equidistant locations is treated differently throughout the DOE Complex. Several sites have developed post-processing routines to approach the requirements of Appendix A to DOE-STD-3009-94. This situation is not ideal because it leaves the calculation of doses to be completed by hand or through spreadsheets. A modest effort should be undertaken to identify the best approach for encoding within MACCS2, possibly as a post-processing option. If this type of option were included as a post-processing option in MACCS2 (to be run prior to running the EARLY module), then the multiple functionality of the EARLY and CHRONC modules would be retained while making dose calculations compliant with the approach recommended by Appendix A of DOE-STD-3009-94.

Source Term Types. The treatment of several source term types important to DOE applications could be improved in MACCS2. Sensible heat algorithms for modeling fire source terms have been implemented for some customers, but systematic treatment of this phenomenology should be standardized in the version of the code available to all DOE users. The current model is limited and may be non-conservative unless combined with a building wake effect model (DOE, 2003f). The code developers could add an option developed by Mills (1987). Additionally, the code does not presently treat deflagration/detonation events accurately. While MACCS2 may not be suitable for mechanistically modeling highly energetic source terms, User's Manual documentation could be expanded to include methods of modeling these events (Steele 1998).

Other user options for treating various aspects of dispersion phenomenology can be explored in future versions of MACCS2. These include plume duration, building wake effects, plume trajectory, puff/plume rise behavior, mixing layer penetration, resuspension, and wet and dry deposition features. While expanded user options would be useful to the DOE consequence analyst, they are not critical to completing current analyses.

The key recommendations for improvements to MACCS2 are summarized in Table 2-2.

Table 2-2 — Summary of Important Recommendations for MACCS2

No.	UI – User Interface Enhancements TM – Technical Model Upgrade	Recommendation
1.	UI	Expand selection of sample problems to include those problem and source term type that are often treated.
2.	UI	Provide Error Diagnostic guidance.
3.	TM	Add DOE-STD-3009-94 Appendix A Post-Processing Algorithm for 95th Percentile, Direction-Independent Doses
4.	UI	Update User interface (planned as part of USNRC program)
5.	TM	Extend sensible heat model to account for area (e.g. pool) releases as well as stack releases with momentum effects.
6.	TM	Consider multiple year option to better sample site data sets that are greater than one year in length

No.	UI – User Interface Enhancements TM – Technical Model Upgrade	Recommendation
7.	TM	Improve close-in model for impacts of building aerodynamic effects, low wind speed conditions
8.	TM	Improve detonation/deflagration (explosive release) approach in code documentation
9.	UI	Provide explicit guidance on major datasets used in DSA applications. The dose conversion factor options should be discussed in greater detail.

2.4 MACCS2 Issues Cited in TECH-25 and Recommended Approaches for Resolutions

Four broad technical issues were explicitly noted in TECH-25 that centered on the MACCS2 software. This section discusses the four main issues and recommended dispositioning.

- Phenomenology: The fire plume model may be non-conservative. It is recommended that the current treatment be carefully used in MACCS2, taking into account building wake effects, sensible energy and spatial dependence of the source term and combustible loading. As a long-term consideration, area source models, such as that proposed by Mills (1987) for pool fire analysis could be made available as a user-specified option in MACCS2.
- Coding Errors: Software defects encountered exercising (1) multiple plume segments and (2) the emergency response model, should be addressed immediately by the code developers. A maintenance version with the major defects corrected should be made available to RSICC. A similar strategy was used for the predecessor software to MACCS2, MACCS, in creating Version 1.5.11.1. In the interim, DOE user guidance should be applied to avoid these conditions in MACCS2 (DOE, 2003f).
- End User Quality Assurance Problem: Dose conversion factors are user-specified data file input options in MACCS2. For example, non-conservative inputs for plutonium radionuclides can be unintentionally selected by users. It is recommended that user instructions (user's manual) address this potential pitfall in running MACCS2. In addition, enhanced training on the options in MACCS2 for dose factor file selection is recommended.
- Poor Documentation: Documentation for MACCS2 should be revised as part of the new software baseline. In particular, the user's guide should provide sample input files for various types of "standard" problem types encountered in both reactor and non-reactor nuclear facility safety analysis.

2.5 Conclusion Regarding Software's Ability to Meet Intended Function

The MACCS2 code 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, MACCS*2 Computer Code Application Guidance for Documented Safety Analysis*, (DOE 2003f), it is judged that it will meet the intended function. Current software concerns and issues can be avoided by understanding MACCS2 limitations and capabilities, and applying the software in the appropriate types of scenarios for which precedents have been identified.

3.0 Lessons Learned

Table 3-1 provides a summary of the lessons learned during the performance of the MACCS2 gap analysis.

Table 3-1 — Lessons Learned

No.	Lesson
1.	Use of NQA-1 or other SQA criteria could not be fully verified. It is obvious that many actions supporting SQA practices have been applied in developing MACCS2, but independent confirmation of the SQA program, practices, and procedures is not possible.
2.	Observance of SQA requirements in the development of safety analysis software such as MACCS2 has not been consistent. It appears to be sporadic in application, poorly funded, and performed as an add-on activity.
3.	While some evidence of pre-development planning is found for early versions of the MACCS2 software, documentation is not maintained as would be expected for compliance with Quality Assurance criteria in Subpart A to 10 CFR 830 (Nuclear Safety Management).
4.	A new software baseline can be produced with "modest" resources (~2 full-time equivalent years) and should be a high priority.
5.	Additional opportunities and venues should be sought for training and user qualification on safety analysis software. This is a long-term deficiency that needs to be addressed for MACCS2 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. Training and Software Improvements (resource estimate) sections follow the ten topical areas.

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. Cross-Reference of Requirements with Subsection and Entry from DOE (2003e)

Subsection (This Report)	Corresponding Entry Table 3-3 from DOE (2003e)	Requirement
4.1	1	Software Classification
4.2	2	SQA Procedures/Plans
4.3	5	Requirements Phase
4.4	6	Design Phase
4.5	7	Implementation Phase
4.6	8	Testing Phase
4.7	9	User Instructions
4.8	10	Acceptance Test
4.9	12	Configuration Control
4.10	13	Error Notification

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 from the RSICC software center (see Table 1-2, under "Documentation Supplied with Code Transmittal"), to make an informed determination of the classification of the software. A user of the MACCS2 software for safety analysis applications would be expected to interpret the information on the software in light of the requirements for dispersion and dose 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. In the software requirements procedure provided by SNL, the MACCS2 software would be deemed Compliance Decision (CD) software SNL (2003).

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	Sufficient information is provided from RSICC and previously transmitted documentation from the software developer. Interpreted in light of Appendix A to DOE-STD-3009-94.

4.1.2 Sources and Method of Review

Documentation supplied with the MACCS2 software package was used along with previously obtained MACCS2 documents as 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).

Due to limited information received from the software developers, extensive use is made of an earlier independent review of the MACCS2 SQA Program (East 1998). The documented review was preceded by an in-depth review at Sandia National Laboratories in 1997. The following, based on the earlier review, provides a good synopsis of the SQA program, prior to and during the period that MACCS2 was developed.

> SNL established a SQA program for Laboratory software in the late 1980s and early 1990s that was compliant with the IEEE Standard for Software Quality Assurance Plans. The final volume was put into place in 1992. The guidelines[3] are documented as shown:
>
> Volume 1 – Software Quality Planning [SNL, 1987]
> Volume 2 – Documentation [SNL, 1995]

[3] - The SNL documentation is clearly described as guidance. The management directing the project may choose not to follow any part, or all, of the recommendations outlined in the guidelines.

Volume 3 – Standard, Practices, and Conventions [SNL, 1986]
Volume 4 – Configuration Management [SNL, 1992]; and
Volume 5 –Tools, Techniques, and Methodologies [SNL, 1989].

The following is a list and description of the necessary documents required for a complete SNL SQA package [SNL, 1986]:

Project Plan: The project plan is a brief overview of the project. It defines the project, describes the organization, proposes schedules and milestones, and defines procedures to ensure the quality of the final product.

Software Requirements Specification (SRSp): The SRSp is a description of the external interfaces and essential requirements of the software in terms of functions, performance, constraints, and attributes. Requirements are objective and measurable. The SRSp is concerned with what is required, not how to achieve it. This document is reviewed by project members, users, and management. They verify that the intent of the SRSp is clear, the software proposed by the SRSp is what is desired, and that the project can proceed to the next development stage.

Design Description: A Design Description documents the design work accomplished during the design phase. Documenting the design prior to coding avoids (or reduces) any design misunderstandings and subsequent re-coding.

Design Review Results: The results of the Design Review are documented in a report, which identifies all deficiencies discovered during the review along with a plan and schedule for corrective actions. The updated design description document, when placed under configuration control, will establish the baseline for subsequent phases of the software life cycle.

Structured Source Code: Implementation is the translation of the detailed design into a computer language; a process commonly called *coding*.

Test Set: The Test Set includes "rich" test data and relevant test procedures and tools to adequately test the application's response to valid as well as invalid data.

Test Set Documentation: The Test Set Documentation (or Software Test Plan) describes the test data, procedures, tools, and overall plan.

Test Results: The results of the tests should be documented to identify all deficiencies discovered.

Maintenance Documentation: Well-documented code and the software design document provide the backbone of maintenance documentation and the starting point for determining training needs.

Training Plan: The preparation of a well thought out training plan is an essential part of bringing a system into smooth operation. If the people, documents, and training techniques are not considered in the early planning for a new system, resources may not be available and training will be haphazard.

User's Manual or Operating Procedures: A user's manual is organized to contain practical information for the individuals required to put the software into action. Depending on the size and type of system, operating procedures may be required as a separate document to cover management of the logical and physical components. Without a properly prepared user's guide or operator instructions, either the time of the user will be wasted determining what to do, or the system will be inappropriately used, or both.

Configuration Management Plan: The Configuration Management Plan lists all modules used by the project, module locations, personnel responsible for controlling changes, and change procedures.

Baseline Table: The Baseline Table lists modules and versions in the project's baselined system.

Change Table: The Change Table lists all changes and enhancements made to the modules. Additional update supporting documents reflect changes and enhancements made to the system.

Of the five SNL software guideline volumes, two[4] were published after the completion of the original MACCS code. The other three[5] were published during the development phase of the MACCS code, but were in place before the beginning of the MACCS2 development.

Although the guidelines were published after the completion of the MACCS code, the MACCS development followed a systematic method in its planning and execution, as did the error reporting and correction. In the initial code development for MACCS2, the same systematic method was followed. It is noted that while draft project, development and test plans were developed and partially implemented with some stages of development, formal approval and implementation was not realized. A draft test plan was followed through MACCS2 Version 1.02 and then apparently abandoned. In summary, the set of SQA plans were never finalized and subsequently, a formal SQA plan was not put into place.

The monthly reports to DOE from SNL and to SNL management from a MACCS2 subcontractor indicated that testing was being performed during the development of the code. However, copies of the testing reports were not available for review at the time of the independent SQA review.

In addition to the testing, SNL contracted the University of New Mexico (UNM) to independently test MACCS2 during development. This testing was published in a draft document [Summa, 1996], but not finalized. The report focused on the following areas:

ATMOS Module: Calculation of the downwind relative air concentration (χ/Q) and of the diffusion parameters by using both the power law and the new look-up table methods

EARLY Module: Calculation of the acute thyroid dose, of the network evacuation centerline dose, of the radial evacuation peak dose, of the crosswind evacuation dose, and the dose when the evacuation speed changes

CHRONC Module: Testing of the ability to turn off the long-term phase and the decontamination model, comparison of intermediate phase and long-term phase doses, and calculation of the intermediate phase dose.

---

4 - The two volumes published after the beginning of the MACCS2 development were the Documentation volume and the Configuration Management volume. The Documentation volume [SNL, 1995] presents a description of documents needed for developing, maintaining, and defining software projects. The Configuration Management volume [SNL, 1992] presents a discussion of configuration management objectives and approaches throughout the software live cycle for software projects at SNL.

5 - The three volumes published before the beginning of the MACCS2 development were Software Quality Planning volume, Standards, Practices, and Conventions volume, and Tools, Techniques, and Methodologies volume. The Software Quality Planning volume [SNL, 1987] presents an overview of procedures designed to ensure software quality. The Standards, Practices, and Conventions volume [SNL, 1986] presents standards and practices for developing and maintaining quality software at SNL and includes a description of the documents needed for a complete SQA package at SNL. The Tools, Techniques, and Methodologies volume [SNL, 1989] presents evaluations and a directory of software tools and methodologies available to SNL personnel.

The testing by UMN was done in an iterative manner. Errors discovered by UNM resulted in coding changes and a new version of the code. The new code version would then be retested by UNM for the function in question. This process would continue until the function worked correctly. However, it is unclear if UNM retested the previous functions that had earlier tested correctly. The UNM testing did not include any of the preprocessors developed by SNL nor did it include the COMIDA (food pathways) module.

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. Because SQA plan and procedures from the software developer were not available, a thorough evaluation was not possible. Based on discussions with previous MACCS2 project leads, the SQA Program reviewer from 1997-1998 (J. East), and East (1998), it is believed that most elements of a compliant SQA plan and procedures were in place and followed. However, definitive confirmation through written, approved documentation is not available.

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
2.1	Verify that procedures/plans for SQA (**SQA Plan**) have identified organizations responsible for performing work; independent reviews, etc.	Possibly. No written confirmation.	Judged that draft program elements were followed – but written confirmation was not obtained.
2.2	Verify that procedures/plans for SQA (**SQA Plan**) have identified software engineering methods.	Possibly. No written confirmation.	Judged that draft procedure/plan elements were followed – but written confirmation was not available.
2.3	Verify that procedures/plans for SQA (**SQA Plan**) have identified documentation to be required as part of program.	Possibly. No written confirmation.	Judged that draft procedure/plan elements were followed – but written confirmation was not available.
2.4	Verify that 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.	Possibly. No written confirmation.	Judged that draft procedure/plan elements were followed – but written confirmation was not available.
2.5	Verify that procedures/plans for SQA (**SQA Plan**) have identified software reviews and schedule.	Possibly. No written confirmation.	Judged that draft procedure/plan elements were followed – but written confirmation was not available.
2.6	Verify that procedures/plans for SQA (**SQA Plan**) have identified methods for error reporting and corrective actions.	Possibly. No written confirmation.	Judged that draft procedure/plan elements were followed – but written confirmation was not

Criterion Number	Criterion Specification	Compliant	Summary Remarks
			available.

4.2.2 Sources and Method of Review

This review was based on Chanin (1997), East (1998) and Summa (1996), and several emails documented as appendices to East (1998).

4.2.3 Software Quality-Related Issues or Concerns

Lack of a verifiable, written set of SQA plan and procedures for MACCS2 should be addressed promptly.

4.2.4 Recommendations

Recommendations related to this topical area are provided as follows:

- Update and finalize draft report by Summa (1996) on *Pre-Release Verification Testing of the MACCS2 Code.*
- Document brief SQA plan for Version 1.12 of MACCS2 (Revise as needed for future updates released to RSICC for public distribution).

4.3 Topical Area 3 Assessment: Requirements Phase

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

Because of limited information received from the software developers, the Requirement Phase topical area could not be evaluated. However, an "incomplete" draft Requirements document has been prepared for MACCS2 (Bixler, 2000). It is likely to need to be completely rewritten to comply with the established set of criteria for this topical area.

4.3.1 Criterion Specification and Results

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.	No.	Draft Requirements Document may exist, but is incomplete and would likely need to be rewritten.
3.2	Software requirements are specified, documented, reviewed and approved.	No.	Draft Requirements Document may exist, but is incomplete and would likely need to be rewritten.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
3.3	Requirements define the functions to be performed by the software and provide detail and information necessary to design the software.	No.	Draft Requirements Document may exist, but is incomplete and would likely need to be rewritten.
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.	No.	Draft Requirements Document may exist, but is incomplete and would likely need to be rewritten.
3.5	Acceptance criteria are established in the software requirements documentation for each of the identified requirements.	No.	Draft Requirements Document may exist, but is incomplete and would likely need to be rewritten.

4.3.2 Sources and Method of Review

This review was based on information contained in East (1998) and Bixler (2000).

4.3.3 Software Quality-Related Issues or Concerns

Lack of a verifiable, written Requirements Document for MACCS2 should be addressed as part of the written SQA Plan and Procedures for this software.

4.3.4 Recommendations

Develop a Requirements Document for MACCS2 that is consistent with the draft developed early in the MACCS2 project but never completed. It should reflect NRC-specified needs for the software as well as those required by DOE and other organizations that sponsored revisions to the software.

4.4 Topical Area 4 Assessment: Design Phase

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

A Software Design Document has not been provided by the MACCS2 software developers. To permit a limited evaluation, an alternative process, that of reviewing model description sections in three reports was applied. The assumption was made that documentation describing earlier versions of MACCS is applicable to MACCS2.

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.	Partial.	Elements of this criterion may be inferred from documentation.
4.2	Code developer 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.	Indeterminate.	-
4.3	The following design should be present and documented: the design should specify the interfaces, overall structure (control and data flow) and the reduction of the overall structure into physical solutions (algorithms, equations, control logic, and data structures).	Partially compliant.	Inferred from MACCS and MACCS2 documentation.
4.4	The following design should be present and documented: that 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.	Partially compliant.	Inferred from documentation.
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 events that can affect the computer program.	Indeterminate.	-
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.	Uncertain.	Some evidence is available of the design intent relating back to requirements.
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.	Partially compliant.	Most of the listed elements are addressed in documentation specified in Section 4.4.2.

Criterion Number	Criterion Specification	Compliant	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.	No.	User knowledge and accident analysis background is required to understand if inputs/outputs are logical.
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.	-
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.	Indeterminate.	It is uncertain whether the software developer has maintained this information.
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.	Partial compliance, incomplete.	Some measure of verification provided in Summa (1996).
4.12	The organization responsible for the design assured that the test results adequately demonstrated the requirements were met.	Uncertain.	-
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.	Yes (1992 – 1995); No (1995 – 1997)	Early MACCS2 project had adequate independence. Second period of effort lacked independence.
4.14	The results of the Independent Review are documented with the identification of the verifier indicated.	Partial.	(Same as above).
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.	-
4.16	Software design documentation was completed prior to finalizing the Independent Review.	Uncertain.	-
4.17	The extent of the IR and the methods chosen are shown to be a function of:	Uncertain.	Insufficient information is available or provided to be

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	the importance to safety, the complexity of the software, the degree of standardization, and the similarity with previously proven software.		able to determine if this criterion was met.

4.4.2 Sources and Method of Review

Design requirements were evaluated through review of the following documents:

- Chanin, 1990, D.I. Chanin, J.L. Sprung, L.T. Ritchie, H-N Jow, and J.A. Rollstin, *MELCOR Accident Consequence Code System (MACCS). Volume 1: User's Guide*; H-N Jow, J.L. Sprung, J.A. Rollstin, L.T. Ritchie, and D.I. Chanin, *Volume 2: Model Description*; J.A. Rollstin, D.I. Chanin, and H-N Jow, *Volume 3: Programmer's Reference Manual*; NUREG/CR-4691, Sandia National Laboratories, published by the U.S. Nuclear Regulatory Commission, Washington, DC, 1990.

- Chanin, 1992a, D. Chanin, J. Rollstin, J. Foster, and L. Miller, *MACCS Version 1.5.11.1: A Maintenance Release of the Code*, Sandia National Laboratories, Albuquerque, NM, July 14, 1992.

- Dobbe 1990, C.A. Dobbe, E.R. Carlson, N.H. Marshall, E.S. Marwil, J.E. Tolli. *Quality Assurance and Verification of the MACCS Code, Version 1.5,* Idaho National Engineering Laboratory, Idaho Falls, ID, NUREG/CR-5376 (EGG-2566)

- Summa, F.J., (1996) and F.E. Haskin. *Pre-Release Verification Testing of the MACCS2 Code.* University of New Mexico, Albuquerque, NM

- Chanin, D., (1997). Software Quality Assurance Procedures Followed with MACCS2, Letter to K. O'Kula (September 1997).

4.4.3 Software Quality-Related Issues or Concerns

A verifiable, written Software Design Document for MACCS2 should have been part of the written SQA Plan and Procedures for this software. Upgrades to the Model Description and other documentation can meet the intent of the Software Design Document for an interim period. However, in reconstituting the baseline for MACCS2, it is highly desirable that a new Software Design Document be developed.

4.4.4 Recommendations

Documenting the software design implemented in MACCS2 is not required at this time. Upgrades to the Model Description and other documentation meet the intent of the Software Design Document for the time being. However, before meeting all prerequisites for the DOE toolbox, a software design report should be prepared.

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.	Yes.	User guide, model description, and code listing from RSICC confirm meeting this criterion.
5.2	Implemented software was analyzed to identify and correct errors.	Uncertain	Not possible to verify.
5.3	The source code finalized during verification (this phase) was placed under configuration control.	Partial.	Likely, but cannot be verified.
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.	Partial.	Copy of software and test case description are available. Not possible to trace to requirements and design documents.

4.5.2 Sources and Method of Review

Documentation listed in Table 1-3 was reviewed to complete review of this criterion. The code listing is available from RSICC upon transmittal of MACCS2 to requesting user groups.

4.5.3 Software Quality-Related Issues or Concerns

Not all criteria can be confirmed due to the lack of written records on implementation. However, based on discussions with project lead for MACCS2 and the subcontractor whom supported the project, it is inferred that most of these requirements were met.

4.5.4 Recommendations

No recommendations related to this topical area are made.

4.6 Topical Area 6 Assessment: Testing Phase

This area corresponds to the requirement entitled Testing Phase in Table 3-3 of DOE (2003e). A Software Test Report has not been provided by the MACCS2 software developers. Instead, a limited evaluation is performed applying Chanin (1997), East (1998), and the related documents listed in Table 1-3 as a basis to address the criteria in Table 4.6-1.

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.	Documentation supports the satisfaction of this criterion.
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 ensured that the software adequately and correctly performed all intended functions.	Indeterminate.	Not able to confirm this criterion.
6.3	Testing demonstrated that the compute program properly handles abnormal conditions and events as well as credible failures	Not certain.	No detailed record is available on outcome of testing for abnormal conditions and credible failures.
6.4	Testing demonstrated that the compute program does not perform adverse unintended functions.	Not certain.	No detailed record is available on outcome of testing for adverse unintended functions.
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.	Uncertain.	Testing report(s) not available so not known how extensive test program was. Current suite of test cases supplied with software include commercial reactor and DOE nuclear facility examples.
6.6	Test Phase documentation includes test procedures or plans and the results of the	Partial compliance.	No detailed record of testing is available. It is known that

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	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.		testing was conducted on MACCS2, and it is judged that the final version (1.12) performs as intended. However, resolution of unsuccessful cases is not possible to check, nor is traceability between test results and software requirements.
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.	Partial in some cases. Uncertain.	No detailed record of test procedures and plans was available. It is believed that this criterion was partially met with respect to: (1), (2), (3), (6), and (9). Complete verification is not possible based on lack of documentation from developer.

4.6.2 Sources and Method of Review

Documentation listed in Table 1-3 was reviewed to complete review of this criterion.

4.6.3 Software Quality-Related Issues or Concerns

Lack of a test report for MACCS2 forces the review to infer test case program results and outcome based on limited information. As was noted previously, the initial period (1992 – 1994) of MACCS2 development had satisfactory procedures and independence during testing. Later testing (1995 – 1997) was not as robust, but did feature an appropriate level of independence in work by the University of New Mexico as an independent checker of changes by SNL (Summa, 1996). It is not possible to verify how complete the University program was, relative to the full software source code package. Apparently, most but not all changes were checked during this phase of the MACCS2 program.

Other testing of the MACCS2 software is encouraged in terms of comparing test output with other, independent results, as listed in Criterion 6.5. (See Recommendations below, Section 4.6.5).

4.6.4 Recommendations

A verifiable, written Test Report Document for MACCS2 should have been part of the written SQA Plan and Procedures for this software. Upgrades to the MACCS2 new software baseline will require that a Test Case Description and Report be completed.

Test cases should include more example types that serve to demonstrate adequacy of MACCS2 software for specific source term types. It is recommended that a standard set of problem types include

deflagration/detonation and fire-related source terms. Observed results and data from experiments, field tests, or specific "known" dispersion results could be compared to test runs made with the MACCS2 software.

4.7 Topical Area 7 Assessment: User Instructions

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

User instructions for MACCS2 and its preprocessor programs have been documented (Chanin, 1997; Chanin, 1998). Considered along with DOE-specific input preparation guidance in DOE (2003e), and the older MACCS model (Chanin, 1990; Chanin, 1992a), there is sufficient information to evaluate compliance to this requirement.

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.	Yes	MACCS and MACCS2 models are described sufficiently.
7.2	User's manual or guide includes approved operating systems (for cases where source code is provided, applicable compilers should be noted).	Yes	RSICC software center distribution notes are available.
7.3	User's manual or guide includes description of the user's interaction with the software.	Yes.	-
7.4	User's manual or guide includes a description of any required training necessary to use the software.	No.	Training requirements are not discussed in MACCS2 documentation.
7.5	User's manual or guide includes input and output specifications.	Yes.	Well documented I/O specifications.
7.6	User's manual or guide includes a description of software and hardware limitations.	Partial.	Some areas in terms of software/hardware limitations are discussed.
7.7	User's manual or guide includes a description of user messages initiated as a result of improper input and how the user can respond.	No.	The user has limited diagnostic assistance to correct errors. MACCS2 documentation does not address error messages satisfactorily.
7.8	User's manual or guide includes information for obtaining user and maintenance support.	Partial.	RSICC-distributed software packages contain email and phone contact information. User interaction with code developers

Criterion Number	Criterion Specification	Compliant	Summary Remarks
			is limited.

4.7.2 Sources and Method of Review

Compliance with this requirement was evaluated by review of documentation listed in Table 1.3.

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:

- User diagnostic assistance during software execution is limited and should be expanded. The User's Guide content is too brief on user-induced software problems. Common errors and warning messages could be included with suggested solutions.
- A simple training set of recommendations would be useful. The novice user could be tasked with two to three simple problem types with output information. The current sample case file could take on this function if prioritized correctly.
- Help and internet/email technical contact information should be provided.
- MACCS2 limitations should be made more explicit in the User's Guide.
- Dose conversion data sets: Specific guidance should be provided in selecting various options for dose conversion factors.

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 then is accepted for use. Much of the testing is the burden of the user organization, but the developing organization assumes 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.	No documentation was received describing the acceptance testing of MACCS2 development.
8.2	To the extent applicable to the developer, acceptance testing was performed prior to approval of the computer program for use.	Uncertain.	No documentation was received describing the acceptance testing of MACCS2 development.
8.3	To the extent applicable to the developer, software validation was performed to ensure that the installed software product satisfies the specified software requirements. The engineering function (i.e., an engineering operation an item is required to perform to meet the component or system design basis) determines the acceptance testing to be performed prior to approval of the computer program for use.	Uncertain.	No documentation was received describing the acceptance testing of MACCS2 development.
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.	Partial	The MACCS2 software package from RSICC includes a series of test case inputs/outputs. These cases serve can be viewed as providing users and user groups with a mechanism for deciding if the MACCS2 software is correctly installed and functioning properly.

4.8.2 Sources and Method of Review

Software package for code transmittal and documentation listed in Table 1.3 were reviewed.

An Acceptance Test protocol was not provided to the gap analysis. There is no known formal procedure to assure that an installed version of MACCS2 is working properly. An Installation and Checkout procedure does not exist for MACCS2 (Bixler, 2000).

4.8.3 Software Quality-Related Issues or Concerns

There are no software quality issues or concerns for this requirement.

4.8.4 Recommendations

No recommendations are made for this topical area.

4.9 Topical Area 9 Assessment: Configuration Control

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

No Software Configuration and Control Document was provided by the software developers. The requirement could not verified as having been met.

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 developers 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	MACCS2 is labeled and documented for release as Version 1.12. However, no documentation was provided to provide detail on how configuration control was achieved and maintained during development.
9.2	Implementing procedures meet applicable criteria for configuration identification, change control and configuration status accounting.	Uncertain	-

4.9.2 Sources and Method of Review

Discussions with previous SNL staff have provided some, but insufficient information on which to evaluate this requirement. It has been indicated that a Configuration Control system was in place during development of MACCS2 (Bixler, 2000). However, no written description of this system and the methods employed to assure configuration control were made available.

4.9.3 Software Quality-Related Issues or Concerns

Lack of a Software Configuration and Control document for MACCS2 forces the review to infer compliance based on limited information. Additionally, discussions with MACCS2 users in the DOE Complex have indicated that several versions may be in existence. This would imply lack of good practice with regard to configuration control.

4.9.4 Recommendations

It is recommended that a full-scope Software Configuration and Control document be issued as part of the new software baseline.

4.10 Topical Area 10 Assessment: Error Impact

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

An Error Notification and Corrective Action document was not transmitted by the SNL software developers. Thus, the evaluation of compliance with this criterion is limited and is based on interpretation of the documents listed in Table 1.3 and from discussions with MACCS2 code staff.

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 problem reporting and corrective action process used by the software developing organization addresses the appropriate requirements of the developing organization's corrective action system, and are documented in implementing procedures.	Uncertain.	The process used for monitoring errors and user feedback on MACCS2 could not be adequately evaluated due to lack of input from the software developer.
10.2	Method(s) for documenting (Error Notification and Corrective Action Report), evaluating, and correcting software problems describe the evaluation process for	Uncertain.	The method(s) used for monitoring errors and user feedback on MACCS2 could not be adequately evaluated

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	determining whether a reported problem is an error.		due to lack of input from the software developer.
10.3	Method(s) for documenting (Error Notification and Corrective Action Report), evaluating, and correcting software problems define the responsibilities for disposition of the problem reports, including notification to the originator of the results of the evaluation.	Uncertain.	-
10.4	When a problem is determined to be an error, then action to document, evaluate and correct, as appropriate, is provided for handling how the error relates to appropriate software engineering elements.	Uncertain.	-
10.5	When a problem is determined to be an error, then action to document, evaluate and correct, as appropriate, is provided for handling how the error impacts past and present use of the computer program	Uncertain.	-
10.6	When a problem is determined to be an error, then action to document, evaluate and correct, as appropriate, is provided for handling how the corrective action impacts previous development activities	Uncertain	-
10.7	When a problem is determined to be an error, then action to document, evaluate and correct, as appropriate, is provided for handling how the users are notified of the identified error, its impact; and how to avoid the error, pending implementation of corrective actions.	Uncertain	-

4.10.2 Sources and Method of Review

Limited documentation was available for this review. SNL has reported that a Software Reporting system was implemented for MACCS2 (Bixler, 2000). However, its effectiveness or timeliness could not be reviewed. Instead, two software defect notifications have been used to infer the approach taken for error/defect reporting and dispositioning.

4.10.3 Software Quality-Related Issues or Concerns

While an error/defect notification process is institutionalized at Sandia National Laboratories, it is not clear how it is effectively used. There appears to be limited use of the reporting system at RSICC.

Known software defects still exist in MACCS2 despite developer awareness and the obvious approach toward correction (DOE, 2003f). The two defects impact results during multiple-plume segment calculations, and in use of the emergency response model. Only the first defect would impact typical calculations supporting Documented Safety Analyses. Nonetheless, both defects should be corrected without additional delay.

4.10.4 Recommendations

As part of the new software baseline for MACCS2, a comprehensive Software Error Notification and Corrective Action Report should be provided. Expanded use of the RSICC user network is also suggested to provide more timely reporting of user issues, software news, suggested strategies for resolving software problems, and general communications.

Known software defects in MACCS2 should be corrected immediately, and a new maintenance version of the software made available to the user community.

4.11 Training Program Assessment

Current MACCS2 training opportunities are limited and not well publicized. Comprehensive training should be provided on a more frequent basis.

The Energy Facility Contractors Group (EFCOG) Workshops suggest two annual opportunities to provide training to the core DOE user group. The winter session is during the Safety Basis Subgroup meeting and the summer session is organized for the larger Safety Analysis Working Group. Multi-day MACCS2 training at these two workshops would potentially reach 300 DOE MACCS2 users, managers, regulators, and oversight groups.

It is also strongly suggested that training be offered for certification. This level of user proficiency could be measured by demonstrating competency through a written exam and software execution of a set of test cases.

4.12 Software Improvements and New Baseline

Software improvements for MACCS2 for a Nuclear Regulatory Commission (NRC)-sponsored program have been documented by Bixler (2000). The new software, WinMACCS, will focus on developing a graphical user interface to MACCS2, its preprocessors, and the related post-processors. For this modification, a slightly modified version of MACCS2 will become a module of WinMACCS. Modifications to the existing MACCS2 for WinMACCS were described as falling in two categories: (1) correcting all known FORTRAN errors/problems; and (2) supporting the interface between the "front" end and the FORTRAN modules.

The NRC-sponsored program, despite user interface improvements, does not address the majority of SQA issues associated with Version 1.12 of MACCS as identified in this report. The minimum remedial program required to yield the new software baseline for MACCS2 was discussed earlier as part of Table 1.1. Included are upgrades to software documents that constitute baseline for software, including:

- 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.

Despite the priority and attention to the user interface, the SNL document provides a reasonable estimate of the level of effort needed to meet an earlier version of ASME NQA-1. The SNL report is used to yield an estimate of the program and level of effort required to upgrade the MACCS2 computer software was prepared by SNL using NP-19 in Bixler (2000). NP-19 was identified earlier, and is a SNL procedural guide that implements an earlier version of Subpart 2.7 to NQA-1, specifically NQA-2a-1990. The minimum set of actions, to be applied to MACCS2 are taken from Bixler (2000) and are:

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

While not exactly matching up with the program proposed here, the SNL proposed program is similar to the requirements outlined in this report. Furthermore, the estimates are based on Sandia National Laboratory resources, and as such, are taken as more accurate resource estimates than could be provided otherwise. The overall SQA upgrade program in the SNL program is estimated to require 1.5 full-time equivalent years to complete. The requirements are matched against the requirements earlier, in this document (Table 4.12-1). The overall level of effort, 1.5 FTE-years is rounded up to 2 FTE-years as the final estimate for resource allocation to perform the upgrades required to compensate for MACCS2's known SQA gaps. The estimate compares favorably with an independent 2-FTE-year value generated for a SQA plan that follows ANSI/ANS-10.4 (WSRC, 1998).

Table 4.12-1. Comparison of SQA Upgrade Steps Discussed in Bixler (2000) with the Approach Discussed in DOE (2003e)

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

5.0 Conclusions

The gap analysis for Version 1.12 of the MACCS2 software, based on a set of requirements and criteria compliant with NQA-1, has been completed. Of the ten SQA requirements for existing software classified as level B (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). Remedial actions are recommended before MACCS2 meets SQA criteria for the remaining eight requirements.

A new software baseline is recommended for MACCS2. Suggested remedial actions for this software would warrant upgrading software documents that describe the new baseline. At minimum, it is recommended that software improvement actions be taken, especially:

1. correcting know defects
2. upgrading user technical support activities
3. providing training on a regular basis, and
4. developing new software documentation.

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.

Additionally, the user's documentation should be augmented to include error diagnostic advice and suggested inputs for prototypic problem types.

Once these actions have been accomplished, MACCS2 version 1.12 is qualified for the Central Registry. Approximately two full-time equivalent years is estimated to complete these actions.

It was determined that the MACCS2 code as it currently stands does meet its intended function for use in supporting documented safety analysis. However, until the remedial program is completed MACCS2 users should be aware of current limitations and capabilities of the software for supporting safety analysis.

6.0 Acronyms and Definitions

ACRONYMS:

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

DEFINITIONS:

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

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.

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

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

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]

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 Analysis Software Group (SASG) — A group of technical experts formed by the Deputy Secretary in October 2000 in response to Technical Report 25 issued by the Defense Nuclear Facilities Safety Board (DNFSB). This group was responsible for determining the safety analysis and instrument and control (I&C) software needs to be fixed or replaced, establishing plans and cost estimates for remedial work, providing recommendations for permanent storage of the software and coordinating with the Nuclear Regulatory Commission on code assessment as appropriate.

Safety-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. [DOE O 414.1B]

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. [DOE O 414.1B]

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 the proper accident analysis of nuclear facilities; the proper analysis and design of safety SSCs; and, the proper identification, maintenance, and operation of safety SSCs. [DOE O 414.1B]

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

Toolbox Codes — A small number of standard computer models (codes) supporting DOE safety analysis, having widespread use, and of appropriate qualification that are maintained, managed, and distributed by a central source. 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].

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 (2003f). MACCS2 Computer Code Application Guidance for Documented Safety Analysis, Interim Report, (September 2003).

Mills (1987).

SNL (1986). *Sandia Software Guidelines: Volume 3: Standards, Practices, and Conventions.* Sandia National Laboratories, Albuquerque, NM, SAND85-2346.

SNL (1987). *Sandia Software Guidelines: Volume 1: Software Quality Planning.* Sandia National Laboratories, Albuquerque, NM, SAND85-2344.

SNL (1989). *Sandia Software Guidelines: Volume 5: Tools, Techniques, and Methodologies.* Sandia National Laboratories, Albuquerque, NM, SAND85-2348.

SNL (1992). *Sandia Software Guidelines: Volume 4: Configuration Management.* Sandia National Laboratories, Albuquerque, NM, SAND85-2347.

SNL (1995). *Sandia Software Guidelines: Volume 2: Documentation.* Sandia National Laboratories, Albuquerque, NM, SAND85-2345.

Summa, F.J., (1996) and F.E. Haskin. *Pre-Release Verification Testing of the MACCS2 Code.* University of New Mexico, Albuquerque, NM.

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. See Appendix A for an example of the input to the table prepared for the MACCS2 code.

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)	
Recommended Documentation - Supplied with Code Transmittal upon Distribution or Otherwise	1. 2. 3. 4.

Table 2. Summary Description of Subject Software	
Type	**Specific Information**
Available	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 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[6]]

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:

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

Requirement 2 – SQA Procedures/Plans (Table 3-2 of SQA Plan/Criteria (DOE, 2003a)
ASME NQA-1 2000 Section 200
IEEE Standard 730, *IEEE Standard for Software Quality Assurance Plans.*
IEEE Standard 730.1, *IEEE Guide for Software Quality Assurance Planning.*

2. Software 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.aProvide examples of problem/error notification to users and the process followed to address the deficiency. Attach files as necessary.

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

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