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

Final Report

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

May 2004

INTENTIONALLY BLANK

FOREWORD

This report documents the outcome of an evaluation of the Software Quality Assurance (SQA) attributes of the radiological dispersion computer code, MACCS2, relative to established software requirements. This evaluation, a "gap analysis", is performed to meet commitment 4.2.1.3 of the Department of Energy's Implementation Plan to resolve SQA issues identified in the Defense Nuclear Facilities Safety Board Recommendation 2002-1.

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

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

INTENTIONALLY BLANK

REVISION STATUS

Page/Section	Revision	Change
1. Entire Document	1. Interim Report	1. Original Issue
2. Entire Document	2. Final Report, May 3, 2004	2. Updated all sections per review comments.

INTENTIONALLY BLANK

CONTENTS

INTENTIONALLY BLANK

TABLES

INTENTIONALLY BLANK

Software Quality Assurance Improvement Plan: MACCS2 Gap Analysis

EXECUTIVE SUMMARY

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

The 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 analysis 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 as contained in U.S. Department of Energy, *Software Quality Assurance Plan and Criteria for the Safety Analysis Toolbox Codes*, (DOE, 2003e). It was determined that MACCS2 code does meet its intended function for use in supporting documented safety analysis. However, as with all safety-related software, users should be aware of current limitations and capabilities of MACCS2 for supporting safety analysis. Informed use of the software can be assisted by the current set of MACCS2 reports (refer to Table 1-3), and the code guidance report for DOE safety analysts, *MACCS2 Computer Code Application Guidance for Documented Safety Analysis*, (DOE, 2004). Furthermore, while SQA improvement actions are recommended for MACCS2, no evidence has been found of programming, logic, or other types of software errors in MACCS2 that have led to non-conservatisms in nuclear facility operations, or in the identification of facility controls.

Of the ten primary 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 an acceptable level, i.e., *Classification* (1) and *User Instructions* (7). A third requirement, *Error Notification and Corrective Action* (10), is partially met. Improvement actions are recommended for MACCS2 to fully meet requirement (10) criteria, and the remaining seven requirements. This evaluation outcome is deemed acceptable because: (1) MACCS2 is used as a tool, and as such its output is applied in safety analysis only after appropriate technical review; (2) User-specified inputs are chosen at a reasonably conservative level of confidence; and (3) Use of MACCS2 is limited to those analytic applications for which the software is intended.

By order of priority, it is recommended that MACCS2 software improvement actions be taken, especially:

1. correct known defects
2. upgrade user technical support activities
3. provide training on a regular basis, and
4. revise software documentation.

Performing these four primary actions should satisfactorily improve the SQA compliance status of MACCS2 relative to the primary evaluation criteria cited in this report.

A new software baseline set of documents is recommended for MACCS2 to demonstrate completion of item 4 (above), revise software documentation. The list of baseline documents for revision includes:

1. Software Quality Assurance Plan
2. Software Model Description, including, but not limited to,
    a. Software Requirements
    b. Software Design
3. User's Manual, including, but not limited to,
    a. User Instructions
    b. Test Case Description and Report
    c. Software Configuration and Control
4. Error Notification and Corrective Action Procedure.

Additionally, user documentation should be augmented to include error diagnostic advice and suggested input files for prototypic nuclear facility safety analysis problem types. Approximately two full-time equivalent years is conservatively estimated to upgrade MACCS2 software to be compliant with NQA-1-based requirements for existing software. While most of this effort is logically to be used by the code developer, independent review of the end products is necessary.

A new version of MACCS2, Version 1.13, has recently been released. It is recommended that this version be evaluated relative to the software improvement and baseline document recommendations, as well as the full set of SQA criteria discussed in this report. If this version is found to be satisfactory, it should replace Version 1.12 as the designated version of the software for the toolbox.

It is recommended that MACCS2 user training for DOE safety analysis applications be conducted formally on, at minimum, an annual basis. Prerequisites for, and core knowledge needed by, the user prior to initiating MACCS2 applications should be documented by the code developer.

Approximately one FTE-month per year would be needed to maintain a web-based error notification and corrective action process for MACCS2 (Section 4.10). However, such a process has not been defined in depth for MACCS2 and the other designated toolbox codes.

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 MACCS2 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 March 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), have been 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.

Early in the SQA evaluation program, it was anticipated that each toolbox code owner would provide input information on the SQA programs, processes, and procedures used to develop their software. However, most of the designated toolbox software, including MACCS2, was developed without complete conformance to software quality standards. Furthermore, many of the software developer organizations cannot confirm that key processes were followed. Therefore, most of the SQA evaluation has been preceded with reconstructing software development processes based on anecdotal evidence and limited, supporting documentation.

For independence reasons, the gap analysis is performed by a SQA evaluator not affiliated with the MACCS2 development program. While independent of the code developer, the SQA evaluators responsible for MACCS2 are knowledgeable in the use of the software for accident analysis applications, and understand current software development standards.

1.3 Uses of the Gap Analysis

The gap analysis provides key information to DOE, code developers, and code users.

DOE obtains the following benefits:

- 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 is provided:

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

DOE safety analysts and code users benefit from:

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

1.4 Scope

The gap 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).

The scope of this review did not include auxiliary software referenced in Section 1.7. Dose conversion factor and food pathway software are included with the transmittal of MACCS2 software from the Radiation Safety Information Computational Center (RSICC).

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 MACCS2 software. The ten areas are those particularly applicable to the software development, specifically: (1) Software Classification, (2) SQA Procedures/Plans, (5) Requirements Phase, (6) Design Phase, (7) Implementation Phase, (8) Testing Phase, (9) User Instructions, (10) Acceptance Test, (12) Configuration Control, and (13) Error Impact. Each area, or requirement, is assessed individually in Section 4.

Requirements 3 (Dedication), 4 (Evaluation), and 14 (Access Control), are not applicable for the software development process, and thus are not evaluated in this review. Requirement 4 (Evaluation) is an outline of the minimum steps to be undertaken in a software review, and is complied with by evaluating the areas listed above. Requirement 11 (Operation and Maintenance) is only partially applicable to software development, and is interpreted to be applicable mostly to the software user organization. Several comments on this requirement are covered as an appendix.

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

Phase	**Procedure**
1. Prerequisites	a. Determine whether sufficient information is provided by the software developer 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 Procedure, 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 Procedure, 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.

An information template was transmitted to the Safety Analysis Software Developers in October 2003 to provide basic information as input to the gap analysis process. The main section of the template is attached as Appendix A to the present report, with an example section and references removed. While no written response to the information template was received from the MACCS2 software developers at Sandia National Laboratories (SNL), three sources of information compensated for the lack of a formal reply:

- **Review by East** – A review of MACCS2 SQA program by East (1998b) for the Department of Energy Office of Defense Programs (now National Nuclear Security Administration) was used to assess the MACCS2 program especially during the 1995 – 1997 period.
- **Site visit by SQA Evaluators** – A site visit was made to SNL on 21 January 2004 to meet with MACCS2 software team. The software information template and the kinds of inputs needed to complete the gap analysis report, preliminary findings and observations, and expected recommendations were covered.
- **Discussions and input from MACCS2 consultant** – The SQA evaluation team contacted the primary consultant to the MACCS and MACCS2 efforts, David Chanin. Although not currently associated with the MACCS2 program, Mr. Chanin supplemented the information obtained from SNL.

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. The software is maintained by Sandia National Laboratories (SNL) and is 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 a multiple, straight-line Gaussian plume, Eulerian model. 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 technical and operational summary of the MACCS2 software is contained in Table 1-2.

The set of MACCS2- and MACCS-specific documents reviewed as part of the gap analysis are listed in Table 1-3. The SNL software developers provided Reference 11 (Proposal to Resolve QA Deficiencies in MACCS2) as part of a Safety Analysis Software Group activity in response to TECH-25. Reference 13 (NP 19-1) was provided to support the current assessment. Other documentation was previously received from SNL or RSICC, or gleaned from the technical literature.

[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 general distribution. 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. Note: MELMACCS (MACCS input generator from MELCOR runs) and CHAIN (Radionuclide progeny) are auxiliary codes, and not available from RSICC. CHAIN was developed by Keith Eckerman at ORNL.
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)[3] 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; Internet: http://www-rsicc.ornl.gov/rsicc.html Contact Nathan Bixler (above) or Jocelyn Mitchell @ NRC for authorization Phone: 301-415-5289 Email: jam@nrc.gov

[3] Recommended procurement route is through N. Bixler/J. Mitchell (see below). Except where noted, items shown here are valid when MACCS2 is obtained through RSICC.

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

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, **I**daho National Engineering and Environmental Laboratory, Idaho Falls, Idaho.
Documentation Supplied with Code Transmittal[4]	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). 6. M. L. Young, "READMAC2.txt" (April 1997).
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.

[4] Transmittal of MACCS2 directly from SNL includes training slides on use of MACCS2.

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, nor can it accept more than one weather spatial location. Like all Gaussian models, MACCS2 is not well suited for modeling dispersion close to the source (less than 100 meters from the source) or long-range dispersion (beyond 15 to 20 miles from the source).[5] Momentum effects of highly energetic releases can be approximated. 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 less than one second on a Pentium 2 or 3 GHZ. Running two source terms and sampling a year of weather data (Sample Problem A) requires approximately times on the order of seconds to minutes.
Computer Hardware Requirements	IBM-compatible 486/DX or Pentium PC with 8 MB of RAM The MACCS2 package files require approximately the following disk space when decompressed: MAC2ZIPA.EXE 6 MB MAC2ZIPB.EXE 4 MB FGR_DCF.EXE 2 MB COMIDA2A.EXE 3 MB IDCF_2.EXE 2 MB DOSFAC_2.EXE 4 MB COMIDA2B.EXE 3 MB. 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 code transmittal package from RSICC, which run successfully in a DOS window of Windows 3.1, Windows 95, Windows NT, and Windows 2000. The programs can also be compiled for those PC operating systems with the Microsoft Powerstation FORTRAN 1.0a compiler. The distributed executables will not run under Windows XP. However, upon request, the code developer will supply executables for Windows XP that were compiled using Compaq FORTRAN 95.
Other Versions Available	MACCS 1.5.11.1 (PC486); MACCS 1.5.11.0 (IBM RISC); Version 1.13 was released in March 2004 to RSICC.

[5] Typical PRA calculations often apply a 1000-mile radius basis.

Table 1-3 — Software Documentation Reviewed for MACCS2

No.	Reference
1.	Chanin, 1998, D. Chanin and M. Young, *Code Manual for MACCS2: Volume 1, User's Guide;*, M. Young, D. Chanin, and V. Banjac, *DOSFAC2 User's Guide*, NUREG/CR-6613, SAND97-0594, May 1998, Sandia National Laboratories, Albuquerque, NM.
2.	Chanin, 1998, D. Chanin and M. Young, *Code Manual for MACCS2, Volume 2, Pre-Processor Codes COMIDA2, FGRDCF, IDCF2*; May 1998, NUREG/CR-6613, SAND97-0594, May 1998, Sandia National Laboratories, Albuquerque, NM.
3.	Young, 1997, M. Young, D. Chanin, and V. Banjac, *DOSFAC2 User's Guide*, NUREG/CR-6613, SAND97-0594, December 1997, Sandia National Laboratories, Albuquerque, NM.
4.	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.
5.	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.
6.	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.
7.	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)
8.	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).
9.	Gregory, 1998, J. Gregory, "Software Defect Notifications" (May 1998).
10.	WSRC, 1998, Westinghouse Savannah River Company*, MACCS Input Guidance for SRS Applications (U)*, WSRC-RP-98-00978, (October 1998).
11.	East, 1998a, J.M. East, *Verification of MACCS2 Peak Dose Output (U)*, WSRC-TR-97-204, Westinghouse Savannah River Company, Aiken, SC (July 1998).

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

No.	Reference
12.	East, 1998b, 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).
13.	LANL, 2001 Draft, Los Alamos National Laboratory, *LANL Guidelines for Performing Atmospheric Dispersion Analysis*, Operational Support Tool 300-00-06H, Los Alamos, NM (June 2001).
14.	Bixler, 2000, N. Bixler, *Proposal to Resolve QA Deficiencies in MACCS2*, Informal Memorandum to D. Chung (DOE/DP), Sandia National Laboratories, Albuquerque, NM (2000).
15.	DOE, 2004, U.S. Department of Energy. *MACCS2 Computer Code Application Guidance for Documented Safety Analysis*, Interim Report, (2004). For latest DOE/EH Central Registry report, go to: http://www.eh.doe.gov/sqa/central_%20registry/MACCS2/maccs2.htm
16.	SNL, 2003, Sandia National Laboratories. *Nuclear Waste Management Procedure*, NP 19-1, *Software Requirements*, Revision 10, Waste Isolation Pilot Plant, (May 2003).
17.	Summa, 1996 Draft, F.J. Summa and F.E. Haskin. *Pre-Release Verification Testing of the MACCS2 Code*. University of New Mexico, Albuquerque, NM. Unpublished report prepared for Sandia National Laboratories (1996).
18.	Chanin, 1997, D. Chanin, *Software Quality Assurance Procedures Followed with MACCS2*, Letter to K. O'Kula (September 1997).
19.	Gregory, 1998, J. Gregory, *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. A third topical quality area, *Error Notification*, partially met criteria, but it and the remaining seven topical quality areas were judged either not wholly compliant with the SQA criteria, and/or lacked documentation to confirm compliance. The eight areas that should be addressed for improvement actions are listed in Section 2.2 (Exceptions to Requirements). 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. The ten criteria evaluated are those predominantly executed by the software developer. However, it is noted that criteria for SQA Procedures/Plan, Testing, Acceptance Test, Configuration Control, and Error Notification also have requirements for the organization implementing the software. These criteria were assessed in the present evaluation only from the code developer perspective.

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

No.	Criterion	Reason Not Met	Remedial Action(s)
1.	SQA Procedures/Plans (Section 4.2)	Earlier versions of MACCS (version 1.5.11.1 and older) followed SNL software engineering guidance. Although initially followed, SNL SQA Plan and Procedures for Version 1.12 of MACCS2 software were not explicitly followed.	As part of the new software baseline, the SQA Plan covering version 1.12 and successor versions of MACCS2 should be addressed as a stand-alone report or as part of another SQA document. Any new SQA procedures that provide prescriptive guidance to the MACCS2 software developers should be made available to a SQA evaluator for confirmatory review. • Document 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 documents for Version 1.12 of MACCS2 software, although filed for a 3 – 4 year period, were not maintained. Consequently the Software Requirements Document	As part of the new software baseline for MACCS2, a concise listing of the software requirements should be documented. This can be reported as a stand-alone Software Requirements report, or as part of another MACCS2-specific document. Specific

No.	Criterion	Reason Not Met	Remedial Action(s)
		was never completed.	MACCS2 requirements need to be documented. Those from MACCS may be added to supplement the MACCS2 information, but are not as critical. In contrast, some MACCS-attributes are no longer present in the code, and it would facilitate understanding of the current code requirements to know which ones have been deleted.
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 incomplete model description and user guidance documents, some of which address MACCS, not MACCS2.	As part of the new software baseline for MACCS2, software design information should be provided. This can be reported as a stand-alone report, or as part of another MACCS2-specific document, such as the model description report.
4.	Implementation Phase (Section 4.5)	Written documentation on implementation of Version 1.12 was not produced for MACCS2.	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 never been published.	A test document was prepared by the University of New Mexico (Summa, 1996), but never approved. As part of the new software baseline for MACCS2, this report should be finalized.
6.	Acceptance Test (Section 4.8)	An Acceptance Test protocol was not provided to the gap analysis. No documentation exists that indicates how the code developers tested the code. 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, and proceduralized in the installation files provided by RSICC or SNL.
7.	Configuration Control (Section 4.9)	A MACCS2 Configuration and Control document was not provided for the gap analysis, despite indication that a configuration control system was in place for MACCS2. Files to support this area were not maintained.	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	While a Software Problem Reporting system is apparently functional at SNL, written documentation should be provided to

No.	Criterion	Reason Not Met	Remedial Action(s)
		documentation was provided.	demonstrate 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 to the code's technical model 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). Essentially the same coding error affects calculations with multiple emergency response assumptions, e.g. Sample Problem A. 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. *SNL has indicated that Version 1.13 of the code addresses known errors, especially the multiple plume and multiple emergency response inputs. Version 1.13 was released in March 2004 to the RSICC software center.*

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. Multiple versions arose, in part, as a result of dissemination of instructions on how users can correct the source-term and emergency-response looping errors, combined with the lack of a code update to RSICC after considerable time had elapsed since discovery of these errors. 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 experience and insights to correctly build a usable input file. A U.S. NRC-sponsored upgrade for MACCS2, MACCS3, will improve this feature by developing a Windows©-based system (Bixler, 2000). The new Windows-GUI based "front end" will preserve the core of the code as a standalone executable that can be run in a MS-DOS command window under various versions of Windows in the same manner as MACCS2. The executable for MACCS3 is designed to be "backward-compatible" and allows the use of input files created for MACCS2.

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 with various approaches, rather than standardizing within a configuration controlled version. A modest effort should be undertaken to identify the best approach for performing this type calculation in MACCS2, and making the option available to all users through SNL or RSICC distribution.

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, 2004). The code developers could add an option developed by Mills (1987). The Mills fire model was implemented in a variant of MACCS2 used by Pantex Plant. SNL may include the new fire model in successors to MACCS2. Additionally, the code

as built was not intended to treat deflagration/detonation events. While MACCS2 may not be suitable for mechanistically modeling highly energetic source terms, User's Manual documentation should be expanded to include methods of modeling these events using Gaussian plume modeling (Steele 1998).

Additionally, MACCS2 documentation would be improved by listing user requirements, i.e. prerequisites to meet before running the software. While a set of problems are provided with transmittal of MACCS2, it would be useful to identify the basic knowledge and skill level necessary to meaningfully run the software.

The user interface should be improved in the area of diagnostics. Currently, software diagnostic messages are difficult to interpret and do not provide the user with sufficient understanding of incorrect input or suggested fixes.

Finally, it is suggested that sample problems be augmented to provide guidance in performing primary types of analyses frequently encountered in safety analysis. Suggested are spill, fire, deflagration, and criticality types of releases.

Other user options for treating various aspects of dispersion phenomenology can be explored in future versions of MACCS2. These include treatment of multi-year meteorological data sets, and expanding model selection for building wake effects, low wind speed conditions, plume trajectory, puff/plume rise behavior, mixing layer penetration, resuspension, and dry deposition. While expanded user options would be useful to the DOE consequence analyst, they are not critical to completing current analyses.

Several dose factor options are now available. MACCS2 documentation should provide discussion on the use of different dose conversion factor data sets and provide several default options. Several issues on the user interface and dose conversion factor file look-up by MACCS2 are outside the scope of this report and shall be addressed in the code guidance report (DOE, 2004).

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

Table 2-2 — Summary of Recommendations for MACCS2

No.	UI – User Interface Enhancements TM – Technical Model Upgrade	Recommendation
1.	TM	Correct software to allow multiple plume calculations
2.		Multiple versions of MACCS2 should be curtailed. Existing special purpose versions should be made available to all users through SNL and/or RSICC.
3.	UI	Update User interface (planned as part of USNRC program) and make available to all users.
4.	TM	Add DOE-STD-3009-94 Appendix A Post-Processing Algorithm for 95th Percentile, Direction-Independent Doses
5.	TM	Extend sensible energy model to account for area releases (e.g. pool) as well as stack releases with momentum effects.
6.	TM	Improve detonation/deflagration (explosive release) approach in code and/or documentation
7.	User Requirements	Document knowledge and skill level for users.
8.	UI	Provide Error Diagnostic guidance.
9.	UI	Expand selection of sample problems to include those problem and source term type that are often treated for DOE nuclear facility safety analysis.
10.	TM	Consider multiple year option to better sample site data sets that are greater than one year in length.
11.	TM	Allow use of multi-year meteorological data sets. Consider expanding model selection for close-in or building wake effects, low wind speed conditions, plume trajectory, puff/plume rise behavior, mixing layer penetration, resuspension, and dry deposition.
12.	UI	Document options for using sets of dose conversion factors (DCFs). Provide default databases with older and more current DCFs.

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

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. (This topic was addressed in Section 2.3).
- 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, 2004).

- 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. (This topic was addressed in Section 2.3).
- 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. (This topic was addressed in Section 2.3).

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 2004) and also utilizing information from documentation available from SNL and other sources (Table 1-3), it is judged that it will meet the intended function.

Current software concerns and issues can be avoided by understanding MACCS2 limitations and capabilities. The software can be applied for modeling those types of scenarios where precedents exist, and there is confidence that alternative analysis or experimental data would adequately confirm the code predictions.

Confidence in MACCS2 to meet its intended function is expected to increase with the release of Version 1.13. The software developer has indicated that Version 1.13 corrects known errors in Version 1.12.

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 characteristic of sound SQA practices have been applied in developing MACCS2, but independent confirmation of the SQA program, practices, and procedures is not possible. The principal reason for this outcome was that key files were not maintained.
2.	Observance of SQA requirements in the development of safety analysis software such as MACCS2 has not been consistent. Previous versions of the code were more consistent with SNL SQA practices. MACCS2 SQA for Version 1.12 lacks a documented record of many of the key processes used to develop the new code. Anecdotal evidence is available, but this type of information is difficult to verify.
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. The main products will be a well-integrated set of code manuals providing the detailed design and model description information consistent with current SQA standards.
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.
6.	MACCS2, as well as some of the other software designated for the DOE Safety Analysis Toolbox, is characterized by its developers as a best-estimate, "research tool", and therefore without the compelling need to be compliant with stringent SQA requirements. The original and current sponsor of the MACCS development effort, the Nuclear Regulatory Commission, did not specify rigid SQA requirements be followed, because PSA applications were (and are) intended for the code rather than as a tool for nuclear "limits". This perspective underestimates the benefit of robust SQA practices used *with*, *not after*, any scientific or engineering software development.

4.0 Detailed Results of the Assessment Process

Ten topical areas, or requirements, are presented in the assessment as listed in Table 4.0-1. Training and Software Improvements sections follow the ten topical areas. Included in the software improvements section is an estimate of the resources required to upgrade MACCS2.

In the tables that follow, the topical areas or requirements are labeled as (1.x, 2.x,…,10.x) with the first value corresponding to the topical area and the second value (x), the sequential table order of each criterion. Four qualitative values shall be used to evaluate whether a specific criterion is met:

- Yes – evidence is available to confirm that the program, practices, and/or procedures followed in developing the version of code satisfy the criterion.
- No – sufficient evidence does not exist to demonstrate that the code meets the criterion
- Partial – some evidence exists that the criterion is met, but has not been finalized or is incomplete
- Uncertain – no basis is available to confirm that the criterion is met.

The overall evaluation for a specific requirement is based on the evaluation of the software against the criteria.

Table 4.0-1 — Cross-Reference of Requirements with Subsection and Entry from DOE (2003e)

Subsection (This Report)	**Corresponding Entry Table 3-3 from DOE (2003e)**	**Requirement**	**ASME NQA-1 2000 Section/Consensus Standards**
4.1	1	Software Classification	ASME NQA-1 2000 Section 200
4.2	2	SQA Procedures/Plans	ASME NQA-1 2000 Section 200; IEEE Std. 730, *IEEE Standard for Software Quality Assurance Plans*
4.3	5	Requirements Phase	ASME NQA-1 2000 Section 401; IEEE Standard 830, *Software Requirements Specifications*
4.4	6	Design Phase	ASME NQA-1 2000 Section 402; IEEE Standard 1016.1, *IEEE Guide for Software Design Descriptions;* IEEE Standard 1016-1998, *IEEE Recommended Practice for Software Design Descriptions*
4.5	7	Implementation Phase	ASME NQA-1 2000 Section 204; IEEE Standard 1016.1, *IEEE Guide for Software Design Descriptions;* IEEE Standard 1016-1998, *IEEE Recommended Practice for Software Design Descriptions*
4.6	8	Testing Phase	ASME NQA-1 2000 Section 404; IEEE Std. 829, *IEEE Standard for Software Test Documentation;*

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

4.1 Topical Area 1 Assessment: Software Classification

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

4.1.1 Criterion Specification and Result

Table 4.1-1 lists the subset of criteria reviewed for this topical area and summarizes the findings. Sufficient documentation is provided with software transmittal 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, Level B (or the equivalent) in the classification hierarchy discussed in DOE (2003e), would be selected. For example, based on the software requirements procedure used by SNL for the Waste Isolation Pilot Plant, 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 documentation from the software developer. Interpreted in light of Appendix A to DOE-STD-3009-94, MACCS2 is used by most sites to assist in making control set decisions. However, it is never used to take action alone, i.e., without consideration of other data or supporting information.

4.1.2 Sources and Method of Review

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

This requirement is met. No recommendations are required at this time to improve compliance with the requirement.

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).

To assess this area, extensive use is made of an earlier independent review of the MACCS2 SQA Program (East 1998b). The documented review was preceded by an in-depth review at Sandia National Laboratories in 1997. The following, based on the 1997 review, provides a synopsis of the SQA program, prior to and during the period that MACCS2 was in development.

> 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, Volume 2, was put into place in 1995. The guidelines[6] are documented as listed:

Volume 1 – Software Quality Planning [SNL, 1987]
Volume 2 – Documentation [SNL, 1995]
Volume 3 – Standards, 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.

---

[6] - The SNL documentation is clearly described as guidance. The management directing a given software project may choose all, some, or none, of the recommendations outlined in the guidelines.

User's Manual or Operating Procedures: A user's manual is organized to contain practical information for individuals required to execute the software. 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[7] were published after the completion of the original MACCS code. The other three[8] were published during the development phase of the MACCS code, but preceded 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, and in the error reporting and correction phase. 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 at some stages of the software development, formal approval and implementation were not achieved. 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.

In this criterion and others that are deficient, the SQA evaluators noted that the MACCS predecessor software development tended to be more compliant relative to SNL guidelines. In contrast, the MACCS2 development was not as intentional in applying and maintaining the SNL guidance. Frequently, the MACCS2 development program was characterized by insufficient review prior to releases of the code that would have precluded software inconsistencies. It is believed that an insufficient level of staffing, especially in the latter portion of the MACCS2 development effort, was the chief cause.

---

[7] - The two volumes published after the start of the MACCS2 development activity were Documentation and Configuration Management volumes. 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 life-cycle for software projects at SNL.

[8] - The three volumes published before the start of the MACCS2 development were Software Quality Planning, Standards, Practices, and Conventions, and Tools, Techniques, and Methodologies volumes. 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.

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 have not been maintained.

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 never 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.

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. Once a program section was accepted as valid, testing of the next area of the model began. Upon completion of the testing effort, the test cases were not rerun to ensure that subsequent modifications had no effect on the validity of previously tested functionality. The code testing was a "work in progress" effort, and the final released version of MACCS2 had differences from the successive code versions tested by UNM. The UNM testing also did not include any of the preprocessors developed by SNL nor did it include the COMIDA (food pathways) module. These factors combined with lack of resources led to the Summa (1996) report not being published.

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 (1998b), it is believed that many elements of a compliant SQA plan and procedures were followed informally, at least in part. 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.	Partial. Documentation was not maintained.	Parts of SNL plan were followed early in the MACCS2 project. Later in the project, schedule and resource constraints led to SQA guidelines not being followed.
2.2	Verify that procedures/plans for SQA (**SQA Plan**) have identified software engineering methods.	No.	Parts of SNL plan were followed early in the MACCS2 project. Later in the project, schedule and resource constraints led to SQA guidelines not being followed.
2.3	Verify that procedures/plans for SQA (**SQA Plan**) have identified documentation to be required as part of program.	Partial.	Parts of SNL plan were followed early in the MACCS2 project. Later in the project, schedule and resource constraints led to SQA guidelines not being followed.
2.4	Verify that 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.	Parts of SNL plan were followed early in the MACCS2 project.
2.5	Verify that procedures/plans for SQA (**SQA Plan**) have identified software reviews and schedule.	Partial. .	Parts of SNL plan were followed early in the MACCS2 project. Later in the project, schedule and resource constraints led to SQA guidelines not being followed.
2.6	Verify that procedures/plans for SQA (**SQA Plan**) have identified methods for error reporting and corrective actions.	Partial.	Parts of SNL plan were followed early in the MACCS2 project. Later in the project, schedule and resource constraints led to SQA guidelines not being followed.

4.2.2 Sources and Method of Review

This review was based on Chanin (1997), East (1998b) and Summa (1996), and several emails documented as appendices to East (1998b). It also includes discussions with SNL code developers and David Chanin. Both discussions occurred in January 2004.

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 for Version 1.12 of MACCS2. A preferred course of action is that future versions of MACCS2 be shown to be compliant with SNL SQA Plan and Procedures, or their equivalent.

4.2.4 Recommendations

The criteria are not or partially met. Thus the requirement is not met. Recommendations related to this topical area are provided as follows:

- Incorporate the draft report by Summa (1996) on *Pre-Release Verification Testing of the MACCS2* Code into a document that describes the evolution of MACCS into MACCS2 and also describes the internal SQA procedures followed during that development effort, which spanned 1992 to 1996, and involved contributions from a beta-test group that spanned the DOE Complex.
- Document brief SQA plan for Version 1.12 of MACCS2. Revise as needed for future updates such as Version 1.13 for public distribution, including both SQA plan and procedures.

4.3 Topical Area 3 Assessment: Requirements Phase

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

While requirements for MACCS2 may have been informally agreed upon, and a draft requirements file maintained, no finalized written documentation exists. Anecdotal evidence suggests that there was an understanding of the requirements that MACCS would meet, but this understanding was not documented.

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.	A series of requirements documents were apparently drafted but no longer exist. A full peer-reviewed report would likely need to be written, especially to capture the changes intended in going from MACCS 1.5.11.1 to MACCS2 1.12.
3.2	Software requirements are specified, documented, reviewed and approved.	No.	A series of requirements documents were apparently drafted but no longer exist. Consequently, a full, comprehensive report does not exist to check whether software requirements are specified, documented, reviewed and approved.
3.3	Requirements define the functions to	No.	A series of requirements documents were apparently drafted but no longer exist.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	be performed by the software and provide detail and information necessary to design the software.		Consequently, a full, comprehensive report does not exist to check whether the software requirements define the functions to be performed by the software and provide detail and information necessary to design the software.
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.	A series of requirement-related files were apparently drafted but were not maintained. Consequently, a full, comprehensive report does not exist to check whether requirements are defined for functionality, performance, design inputs, design constraints, installation considerations, operating systems (if applicable), and external interfaces necessary to design the software.
3.5	Acceptance criteria are established in the software requirements documentation for each of the identified requirements.	No.	Memos and e-mails were written on acceptance criteria. However, these were not maintained. Consequently, a full, comprehensive report does not exist to check whether acceptance criteria are established in the software requirements documentation for each of the identified requirements.

4.3.2 Sources and Method of Review

This review was based on information contained in East (1998b) and Bixler (2000). It also includes discussions with SNL code developers and David Chanin. Both discussions occurred in January 2004.

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

The five criteria are not met. Thus the requirement is not met. It is recommended to develop a Requirements Document for MACCS2 that is consistent with the draft information developed early in the MACCS2 project but never completed in a formal report. 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. Especially important are descriptions of the software changes that were made in going from MACCS to MACCS2, including the architectural changes. It would be useful to capture this information in Version 1.13 documentation.

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 was not written for MACCS2. To permit a limited evaluation, an alternative process, that of reviewing model description sections in four reports was applied (Chanin, 1990 Volumes 1 and 2; Chanin, 1992a, and Chanin, 1998). The key assumption was made that documentation describing earlier MACCS development could be extended *partially* 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	A software design was developed, documented, reviewed and controlled.	Partial.	Elements of this criterion may be inferred from documentation, a formal design document was not prepared.
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.	Partial.	Design may be inferred from the final software product, but requirements documentation was not maintained.
4.3	The following design elements 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).	Partial.	Inferred in part from MACCS and MACCS2 documentation. However, the documentation for MACCS in many areas does not reflect what was implemented in MACCS2.
4.4	The following design elements 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.	Partial.	Documentation and discussions imply that operating environment was considered.
4.5	The following design elements 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.	Not applicable to non-process, instrumentation and control software.	-
4.6	A Software Design Document, or	Partial.	Evidence of the design intent

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	equivalent, is available and contains a description of the major components of the software design as they relate to the software requirements.		relating back to requirements may be inferred from MACCS and MACCS2 documents.
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.	Partial.	Most of the listed elements are addressed in documentation listed in Section 4.4.2. Note that the basic code features preserved in MACCS2 (from MACCS) as well as the code changes introduced in MACCS2 are described in the MACCS2 User's Guide (Chanin, 1998).
4.8	A Software Design Document, or equivalent, is available and contains a description of the allowable or prescribed ranges for inputs and outputs.	Partial.	Allowed ranges of input parameters are covered in Chanin (1998). User knowledge and accident analysis background is required to understand if 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.	The required detail to establish compliance is not available.
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.	Uncertain.	It is uncertain whether the software developer has maintained this information during development of MACCS2. User implementation of MACCS2 is guided by a loading and execution sequence provided as part of a "read me" file.
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.	Some measure of verification provided in the draft report by the University of New Mexico (Summa, 1996).
4.12	The organization responsible for the design assured that the test results adequately demonstrated the requirements were met.	Uncertain.	Not possible to verify this area because no documentation is available.

Criterion Number	Criterion Specification	Compliant	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.	Partial: 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.	A portion of the independent review was documented but never published.
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.	Discussions with code staff suggest that this technique was used in the early (1992-1995) program, but lapsed in the later period (1995 – 1997).
4.16	Software design documentation was completed prior to finalizing the Independent Review.	Uncertain.	Not likely, but cannot be confirmed.
4.17	The extent of the Independent Review and the methods chosen are shown to be a function of: the importance to safety, the complexity of the software, the degree of standardization, and the similarity with previously proven software.	Uncertain.	Insufficient information is available to demonstrate that a graded application was followed.

4.4.2 Sources and Method of Review

Design requirements information was evaluated through discussions with David Chanin and the current MACCS2 staff, and review of the reports listed in Table 1-3. In particular, the following were used:

- 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., (Draft-1996) and F.E. Haskin. *Pre-Release Verification Testing of the MACCS2 Code*. University of New Mexico, Albuquerque, NM. Unpublished report prepared for Sandia National Laboratories.

- 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 comprehensive Software Design Document for MACCS2 should have been part of the written SQA Plan and Procedures for this software. A SQA assessment must rely on inferences made from the MACCS2 User's Manual (Chanin, 1998).

4.4.4 Recommendations

Of the seventeen criteria evaluated for this requirement, two (2) are not met, nine (9) are partially met, five (5) are uncertain, and one (1) is not applicable. Thus the requirement is not met. Documenting the software design implemented in MACCS2 Version 1.12 is not required at this time. Upgrades to the Model Description and other documentation can meet the intent of the Software Design Document, but only if the discussion is comprehensive and detailed and provides insights in design differences in transitioning from MACCS to MACCS2. In the long-term, software design information is recommended for the version of MACCS2 ultimately maintained in the 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.	Yes.	User guide, model description, and code listing from RSICC confirm that the code developer met this criterion.
5.2	Implemented software was analyzed to identify and correct errors.	Uncertain.	Not possible to verify due to lack of written documentation. Discussions with code consultant suggest criterion was likely to have been met.
5.3	The source code finalized during verification (this phase) was placed under configuration control.	Uncertain.	Not possible to verify due to lack of written documentation. Discussions with code consultant suggest criterion was likely to have been met.
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. However, 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 MACCS2 developer notes that an Implementation Document does not currently exist for MACCS2.

The code listing is available from RSICC upon transmittal of MACCS2 to 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, at least partially.

4.5.4 Recommendations

Of the four criteria evaluated for this requirement, one (1) is met, two (2) are uncertain, and one (1) is partially met. Thus the requirement is not met. However, it is not recommended that improvements be made related to this topical area for Version 1.12 of MACCS2. Instead, indication of the Implementation processes should be maintained on file for newer versions of the software.

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 based on review of Chanin (1997), East (1998a), East (1998b), 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.	Uncertain.	Not able to confirm this criterion.
6.3	Testing demonstrated that the computer program properly handles abnormal conditions and events as well as credible failures	Uncertain.	No detailed record is available on outcome of testing for abnormal conditions and credible failures.
6.4	Testing demonstrated that the computer program does not perform adverse unintended functions.	Uncertain.	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 include: (1) analysis without computer assistance, e.g. hand or spreadsheet calculations; (2) other validated computer programs; (3) experiments and tests; (4) standard problems with known solutions; (5) confirmed published data and	Uncertain.	Test report(s) are not available so the extent of the in-house test program at SNL is not known. No evidence exists of comparisons of MACCS2 to other independent test results. Current suite of test cases supplied with software for user purposes include commercial reactor and DOE nuclear facility examples.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	correlations.		
6.6	Test Phase documentation includes test procedures or plans and the results of the execution of test cases. The test results documentation demonstrates successful completion of all test cases or the resolution of unsuccessful test cases and provides direct traceability between the test results and specified software requirements.	Partial.	No detailed record of testing is available. It is known that 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, or 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

Discussions with the current MACCS2 staff and the MACCS2 consultant were conducted in January 2004 to supplement information gleaned from the documentation listed in Table 1-3 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. It is noted by East (1998a) that of the enhancements and new output types provided in MACCS2, all but the sector independent and sector dependent peak doses for any organ were tested and verified in the Summa report. Both features were found implemented correctly in MACCS2 (East, 1998a).

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

Of the seven criteria evaluated for this requirement, one (1) is met, four (4) are uncertain, and two (2) are partially met. Thus the requirement is not met. A verifiable, written Test Report Document for MACCS2 should have been part of the written SQA Plan and Procedures for Version 1.12 of this software. Upgrades to the MACCS2 new software baseline will require that a Test Case Description and Report be completed, particularly if Version 1.13 is designated for the toolbox instead of Version 1.12. In addition, review and approval of the test report (Summa, 1996) prepared by the University of New Mexico is strongly recommended

In terms of user implementation, test cases should include example types that serve to demonstrate adequacy of MACCS2 software for specific source term types typically encountered for DOE Documented Safety Analysis. 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 made by the code developer and stored on the MACCS2 CD sent to users.

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.	User's Manual discusses this aspect to a limited extent.
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.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
7.5	User's manual or guide includes input and output specifications.	Yes.	Well documented input specifications. Output discussion should be expanded.
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. This aspect of the MACCS2 document should be clearer in terms of regimes of applicability.
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.	Partial.	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 developer at SNL 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, and from discussions with MACCS2 consultant (Chanin) and with current MACCS2 SNL staff.

4.7.3 Software Quality-Related Issues or Concerns

User instruction documentation is, on the whole, good. No substantive issues or concerns have surfaced.

4.7.4 Recommendations

Of the eight criteria evaluated for this requirement, four (4) are met, one (1) is not met, and three (3) are partially met. The one negative criterion is more than compensated for by the four other positive criteria, and coupled with sufficient written information on which to base the evaluation of this criterion, the requirement is evaluated as met.

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 list of prerequisites for perspective MACCS2 users should be listed in a MACCS2 website or in the code documentation.
- A simple training set of recommendations would be useful. The input files to start with and the output files that one should see upon successful execution could be placed on a MACCS2 website. The novice user could be tasked with two to three simple problem types to set up. The

associated output files would allow a self-check on the code runs. The current sample case file could take on this function if prioritized correctly.
- Help and internet/email technical contact information for SNL should be provided.
- MACCS2 limitations should be made more explicit in the User's Guide.
- 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.

During development of the software, the developing organization is responsible for documenting its procedures and acceptance tests it uses. Once the software is released, user organizations need a test protocol to determine if the software is correctly installed. Implementation for this type of acceptance testing is the responsibility of the user organization.

The criteria below were applied to the developing organization's processes.

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., the 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	Partial	The MACCS2 software package from RSICC includes

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	cases for system installation and integration, user instructions (Refer to Requirement 7 above), and documentation of the acceptance of the software for operational use.		a series of test case inputs/outputs. These cases 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

Acceptance Test information should have been provided by the software developers to assure users that the distributed version of MACCS2 met its testing criteria.

4.8.4 Recommendations

Of the four criteria evaluated for this requirement, three (3) are uncertain, and one (1) is partially met. Thus the requirement is not met.

Documentation, either as a separate report or as part of another SQA report, is needed to describe the testing used for Version 1.12 or successor versions of the MACCS2 software. In addition, the developer is recommended to provide an acceptance test protocol to allow users to determine that MACCS2 is properly installed.

Through discussions with the MACCS2 consultant in January 2004, it was learned that some of the requirements for the Operation and Maintenance area were performed in the development of MACCS2. Although evaluation of this area is not included formally in the ten criteria for this gap analysis, the information is included as Appendix B to this report.

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 was assessed mostly through discussions with the current code developer and the MACCS2 consultant.

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.	No.	No implementing procedures were identified.

4.9.2 Sources and Method of Review

Discussions with previous SNL staff and with the consultant at the time changes were made to MACCS2 have provided limited 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 forced the review to infer compliance based on limited information from discussions. This would imply lack of good protocol with regard to configuration control practices.

Additionally, discussions with MACCS2 users in the DOE Complex have indicated that several versions may be in existence. For example, a version of MACCS2 at Pantex, MAX2_MHC, was developed by SNL to allow use of a different fire plume model and allows the user to input five years of meteorology in

a given run. A NRC-sponsored version, WinMACCS, is being developed for NRC-specific applications. While not violating software practices, multiple versions could cause confusion to users on the MACCS2 version recommended for DSA applications.

4.9.4 Recommendations

Of the two criteria evaluated for this requirement, one (1) is uncertain, and one (1) is not met. Thus the requirement is not met. It is recommended that a full-scope Software Configuration and Control document be issued as part of the new software baseline for the version of MACCS2 that is designated for the DOE software toolbox. Variants of MACCS2 that are used in the DOE Complex should be identified to the full user community with the distinction between the base software and the variants made clear.

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, but an SNL protocol for dealing with software issues has been followed during and after release of Version 1.12 of MACCS2. 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 and the MACCS2 consultant associated with SNL at the time of MACCS2 development.

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.	Partial.	The process used for monitoring errors and user feedback on MACCS2 was inferred from past practice at SNL but limited documentation was made available to the SQA evaluation.
10.2	Method(s) for documenting (Error Notification and Corrective Action Report), evaluating, and correcting software problems describe the evaluation process for determining whether a reported problem is an error.	Yes.	The method(s) used for evaluating and correcting MACCS2 problems were not provided in a written document. The evaluation process followed was inferred.
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.	Partial.	An overall error notification and corrective action process document was not available. Inference drawn from a SNL Software Defect Notice was used to determine that some organizational process must

Criterion Number	Criterion Specification	Compliant	Summary Remarks
			have been followed.
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.	Yes.	A written procedure on the action to address how the potential software error relates to software engineering elements was not available. Inference drawn from two SNL Software Defect Notices was used to determine that some organizational process must have been followed.
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	Partial.	A written procedure on the action to address how the potential software error impacts past and present use of the computer program was not available. Inference drawn from two SNL Software Defect Notices was used to determine that some organizational process must have been followed.
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	Yes.	A written procedure on the action to address how the corrective action impacts previous development activities was not available. Inference drawn from two SNL Software Defect Notices was used to determine that some organizational process must have been followed.
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.	Partial.	A written procedure on the action to address handling how users are notified of identified errors, impacts, error avoidance, and pending implementation of corrective actions was not available. Inference drawn from two SNL Software Defect Notices was used to determine that some organizational process must have been followed.

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 could not be reviewed directly. 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. While software defects in MACCS2 were known beginning in 1997, it took over a year for software defect notices to be disseminated (Gregory, 1998). Thus, the efficacy of the current system of developer – user notification and communication of planned corrective actions is uncertain.

Known software defects still exist in MACCS2 despite developer awareness and the obvious approach toward correction (Bixler, 2000). 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

Of the seven criteria evaluated for this requirement, three (3) are met, and four (4) are partially met. However, several examples of an Error Notification and Corrective Action process have been confirmed by written document that demonstrate a satisfactory process exists. Therefore, it is judged that the overall requirement is partially met.

As part of the new software baseline for MACCS2, a comprehensive Software Error Notification and Corrective Action process 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. In the future, communication of software concerns among MACCS2 users, and the corrective actions to be taken, need to be executed in a timely manner.

Known software defects in MACCS2 should be corrected immediately, and a new maintenance version of the software made available to the user community. Discussions with the staff at SNL in January 2004 indicate that identified MACCS2 software flaws are corrected in Version 1.13. This Version has been released immediately prior to this gap analysis report.

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 provide 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 full Safety Analysis Working Group. Multi-day MACCS2 training at these two workshops would potentially reach 300 DOE MACCS2 users, managers,

regulators, and oversight groups. Site-specific training is also suggested, similar to training conducted by other designated toolbox software developers.

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.

MACCS2 Version 1.13 has recently been released. While the new version corrects known software errors identified since the late 1990s, other improvements made in the software are not known at this time.

The NRC-sponsored WinMACCS version, 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 creation of, or upgrades to, software documents that can then constitute the new baseline for the software, including:

1. Software Quality Assurance Plan
2. Software Model Description, including, but not limited to,
    a. Software Requirements
    b. Software Design
3. User's Manual, including, but not limited to,
    a. User Instructions
    b. Test Case Description and Report
    c. Software Configuration and Control
4. Error Notification and Corrective Action Process.

As is noted in the above list, many of the functions of these documents can be met in the same report. For example, much of the intent of the requirements and design documents can be captured in a comprehensive Model Description report. Furthermore, Document 4 (above) can be a reference to a SNL-wide report as long as the error notification and corrective action process followed for MACCS2 is addressed. The key point of this remedial SQA document list is to provide a comprehensive and cohesive set of documentation that describes the migration path from MACCS to MACCS2, including the architectural changes that were made. In addition, a model description is needed that fully reflects the applicable models in MACCS2, and indicates those MACCS models that are no longer active. Reference to earlier MACCS2- and MACCS-related documents is encouraged, as long as the document in question is still applicable to the version of MACCS2 being described. A comprehensive documentation effort of this level will greatly enhance user understanding of the MACCS2 models.

Despite the priority and attention to the user interface, Bixler (2000) provides a reasonable estimate of the level of effort needed to meet an earlier version of ASME NQA-1. The estimate of the program and level of effort required to upgrade the MACCS2 computer software was prepared based on NP-19 for WIPP applications. 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 described in Bixler (2000) includes:

- Create a Primitive Baseline (PB) document to establish the SQA status of the existing code
- Establish a Verification and Validation Plan (VVP) based on the above
- 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 improvement actions 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 (East, 1998b).

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

ASME NQA-1-2000 Requirements	**Bixler (2000)**	**DOE (2003e)**
	SNL NP 19-1	**Level B Existing Software***
Software Classification		4.1
SQA Procedures/Plans		4.2
Dedication		-
Evaluation	PB	-
Requirements	SRD	4.3
Design		4.4
Implementation		4.5
Testing	VVP, VD	4.6
User Instructions	ID, UM	4.7
Acceptance Test	I&C	4.8
Operation and Maintenance		Appendix B
Configuration Control	CC	4.9
Error Impact	SPR	4.10
Access Control		-

*** Section covered in this report.**

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 primary 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 an acceptable level, i.e., *Classification* (1) and *User Instructions* (7). A third requirement, *Error Notification and Corrective Action* (10), is partially met. Improvement actions are recommended for MACCS2 to fully meet requirement (10) criteria, and the remaining seven requirements. This evaluation outcome is deemed acceptable because: (1) MACCS2 is used as a tool, and as such its output is applied in safety analysis only after appropriate technical review; (2) User-specified inputs are chosen at a reasonably conservative level of confidence; and (3) Use of MACCS2 is limited to those analytic applications for which the software is intended.

It was determined that MACCS2 code does meet its intended function for use in supporting documented safety analysis. However, as with all safety-related software, users should be aware of current limitations and capabilities of MACCS2 for supporting safety analysis. Informed use of the software can be assisted by the current set of MACCS2 reports (refer to Table 1-3), and the code guidance report for DOE safety analysts, *MACCS2 Computer Code Application Guidance for Documented Safety Analysis*, (DOE, 2004). Furthermore, while SQA improvement actions are recommended for MACCS2, no evidence has been found of programming, logic, or other types of software errors in MACCS2 that have led to non-conservatisms in nuclear facility operations, or in the identification of facility controls.

By order of priority, it is recommended that MACCS2 software improvement actions be taken, especially:

- correcting know defects
- upgrading user technical support activities
- providing training on a regular basis, and
- revising software documentation.

Performing these four primary actions should satisfactorily improve the SQA compliance status of MACCS2 relative to the primary evaluation criteria cited in this report.

A new software baseline set of documents is recommended for MACCS2 to demonstrate completion of the revision to software documentation item (above). The list of baseline documents for revision includes:

1. Software Quality Assurance Plan
2. Software Model Description, including, but not limited to,
    a. Software Requirements
    b. Software Design
3. User's Manual, including, but not limited to
    a. User Instructions
    b. Test Case Description
    c. Software Configuration and Control
4. Error Notification and Corrective Action Process.

Additionally, user documentation should be augmented to include error diagnostic advice and suggested input files for prototypic nuclear facility safety analysis problem types. Approximately two full-time equivalent years (2 FTEs) is estimated to complete these actions. Of this level of effort, 1.5 FTE is

estimated for the current software owner, Sandia National Laboratories, and roughly 0.5 FTE is estimated to be required for independent review.

A new version of MACCS2, Version 1.13, has recently been released. It is recommended that this version be evaluated relative to the software improvement and baseline recommendations, and the complete set of SQA criteria discussed in this report. If this version is found to be satisfactory, it should replace Version 1.12 as the designated version of the software for the DOE Safety Software Toolbox.

Currently, MACCS2 training is occasionally offered on an informal basis, and user requirements are not discussed in the code developer's documentation. It is recommended that user training for DOE safety analysis applications be conducted formally on at minimum, an annual basis. Prerequisites for, and core knowledge needed by, the user prior to initiating MACCS2 applications should be documented by the code developer.

Approximately one FTE-month per year would be needed to maintain a web-based error notification and corrective action process for MACCS2 (Section 4.10). However, such a process has not been defined in depth for MACCS2 and the other designated toolbox codes.

6.0 Acronyms and Definitions

ACRONYMS:

AEC	Atomic Energy Commission
ALOHA	Areal Locations of Hazardous Atmospheres (designated toolbox software)
ANS	American Nuclear Society
ANSI	American National Standards Institute
ASME	American Society of Mechanical Engineers
CCPS	Center for Chemical Process Safety
CD	Compliance Decision
CFAST	Consolidated Fire and Smoke Transport Model (designated toolbox software)
CFR	Code of Federal Regulations
DCF	Dose Conversion Factor
DNFSB	Defense Nuclear Facilities Safety Board
DoD	Department of Defense
DOE	Department of Energy
DSA	Documented Safety Analysis
EFCOG	Energy Facility Contractors Group
EH	DOE Office of Environment, Safety and Health
EIA	Electronic Industries Alliance
EM	DOE Office of Environmental Management
EPIcode	Emergency Prediction Information code (designated toolbox software)
EPRI	Electric Power Research Institute
FTE	Full-time equivalent
GENII	Generalized Environmental Radiation Dosimetry Software System - Hanford Dosimetry System (Generation II) (designated toolbox software)
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IP	Implementation Plan
ISO	International Organization for Standardization
MACCS2	MELCOR Accident Consequence Code System 2 (designated toolbox software)
MELCOR	Methods for Estimation of Leakages and Consequences of Releases (designated toolbox software)
NNSA	National Nuclear Security Administration
NRC	Nuclear Regulatory Commission
OCRWM	Office of Civilian Radioactive Waste Management
PSA	Probabilistic Safety Analysis (or Assessment)
QAP	Quality Assurance Program (alternatively, Plan)
RSICC	Radiation Safety Information Computational Center
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.

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

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.

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).

Dedication (of Software) — The evaluation of software not developed under utilizing organization existing quality assurance 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.

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, 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]

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

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]

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]

Software design requirements 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:

Software design
Implementation
Test, and sometimes:
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]

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 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]

Toolbox Codes — A small number of standard computer models (codes) supporting DOE safety analysis, having widespread use, and of appropriate qualification that are maintained, managed, and distributed by a central source. Toolbox codes meet minimum quality assurance criteria. They may be applied to support 10 CFR 830 DSAs provided the application domain and input parameters are valid. In addition to public domain software, commercial or proprietary software may also be considered. In addition to safety analysis software, design codes may also be included if there is a benefit to maintain centralized control of the codes [modified from DOE N 411.1].

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

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

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

7.0 References

Note: The references listed below may not have been used directly in the gap analysis. However, they were used to provide a context for performing the overall code evaluation.

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

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

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

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

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

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

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

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

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

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

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

DOE, U.S. Department of Energy (2004). *MACCS2 Computer Code Application Guidance for Documented Safety Analysis*, (May 2004).

Mills, M.T. (1987). "Modeling the Release and Dispersion of Toxic Combustion Products from Chemical Fires," paper presented at the International Conference on Vapor Cloud Modeling, November 2–4, 1987, Cambridge, MA. (Obtained from the author at ENSR Consulting and Engineering, Acton, MA.)

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.

Appendices

Appendix	Subject
A	Software Information Template
B	Operation and Maintenance Criterion

APPENDIX A. SOFTWARE INFORMATION TEMPLATE

The following is a condensed version of the information request sent to the MACCS2 code developer in October 2003.

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 Available	1. 2. 3. 4. 5.
Input Data/Parameter Requirements	

Table 2. Summary Description of Subject Software	
Type	**Specific Information**
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[9]]

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
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]

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

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		

Type	Description	Frequency of training

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 correctio
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.

APPENDIX B. OPERATION AND MAINTENANCE CRITERION

B.1 Topical Area Assessment: Operation and Maintenance Phase

This area corresponds to the requirement entitled Operation and Maintenance in Table 3-3 of DOE (2003e).

B.1.1 Criterion Specification and Results

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

Table B-1 — Subset of Criteria for Operations and Maintenance Topic and Results

Criterion Number	Criterion Specification	Compliant	Summary Remarks
B.1	During this phase, software shall be controlled to remove latent errors (corrective maintenance), to respond to new or revised requirements (enhancement), or to adapt the software to changes in the operating environment (adaptive maintenance). Software modifications shall be approved, documented, verified and validated, and controlled in accordance with the related life cycle phases.	No.	A written procedure for corrective and adaptive maintenance was not available.
B.2	The validation of modifications shall be subject to selective regression testing to detect errors introduced during the modification of software or operating system components to verify that the modifications have not caused unintended adverse effects and to verify that the modified software still meets its specified requirements.	Partial.	Discussions with MACCS2 consultant indicate that some regression testing was performed. It is not clear if this activity was performed sporadically in the second phase of MACCS2 development or followed a regular schedule.
B.3	Test cases shall be developed and documented to permit confirmation of acceptable performance of the software in the environment in which the software is used. Test cases shall be run whenever the software is installed on a different computer, or when significant hardware or operating system configuration changes are made.	Partial.	Test cases were formally drawn up for Version 1.12 of MACCS2.
B.4	Periodic in-use manual or automatic self-check in-use tests shall be prescribed and performed for those	N/A.	This evaluation criterion was interpreted to be applicable mostly for process control software. It was not

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	computer programs where computer program errors, data errors, computer hardware failures, or instrument drift can affect required performance.		addressed here due to the safety analysis nature of the MACCS2 software.

B.1.2 Sources and Method of Review

This review was based on information contained in East (1998b) and Bixler (2000). It also includes discussions with SNL code developers and David Chanin. Both discussions occurred in January 2004.

B.1.3 Software Quality-Related Issues or Concerns

While some maintenance activities such as configuration control and a software reporting system were in place during the development of MACCS2 Version 1.12, other activities falling under Operations and Maintenance appear to have been performed. However, there is no written record for confirmation purposes.

B.1.4 Recommendations

Of the four criteria evaluated for this requirement, one (1) is not met, two (2) are partially met, and one is judged as not applicable. Thus the requirement is not met.

It is advised that MACCS2 developer consider Operations and Maintenance processes as a tool to check software change effects. Specifically, a process should describe removal of latent errors (corrective maintenance) and adaptation of the software to changes in the operating environment (adaptive maintenance). Regression testing procedures should be developed to ensure that software modifications do not introduce unintended adverse effects. Test case recommendations are made in the body of this report.