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

Software Quality Assurance Improvement Plan: MELCOR 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 MELCOR computer code for leak path factor applications, 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

Section **Page**

FOREWORD **iii**

REVISION STATUS **v**

EXECUTIVE SUMMARY **xiii**

1.0 Introduction **1-1**

1.1 BACKGROUND: OVERVIEW OF DESIGNATED TOOLBOX SOFTWARE IN THE CONTEXT OF 10 CFR 830 1-1
1.2 EVALUATION OF TOOLBOX CODES 1-2
1.3 USES OF THE GAP ANALYSIS 1-2
1.4 SCOPE 1-2
1.5 PURPOSE 1-3
1.6 METHODOLOGY FOR GAP ANALYSIS 1-3
1.7 SUMMARY DESCRIPTION OF SOFTWARE BEING REVIEWED 1-5

2.0 Assessment Summary Results **2-1**

2.1 CRITERIA MET 2-1
2.2 EXCEPTIONS TO REQUIREMENTS 2-1
2.3 AREAS NEEDING IMPROVEMENT 2-2
2.4 CONCLUSION REGARDING SOFTWARE'S ABILITY TO MEET INTENDED FUNCTION 2-3

3.0 Lessons Learned **3-3**

4.0 Detailed Results of the Assessment Process **4-1**

4.1 TOPICAL AREA 1 ASSESSMENT: SOFTWARE CLASSIFICATION 4-2
4.1.1 Criterion Specification and Result 4-2
4.1.2 Sources and Method of Review 4-3
4.1.3 Software Quality-Related Issues or Concerns 4-3
4.1.4 Recommendations 4-3
4.2 TOPICAL AREA 2 ASSESSMENT: SQA PROCEDURES AND PLANS 4-3
4.2.1 Criterion Specification and Result 4-5
4.2.2 Sources and Method of Review 4-6
4.2.3 Software Quality-Related Issues or Concerns 4-7
4.2.4 Recommendations 4-7
4.3 TOPICAL AREA 3 ASSESSMENT: REQUIREMENTS PHASE 4-7
4.3.1 Criterion Specification and Results 4-8
4.3.2 Sources and Method of Review 4-8
4.3.3 Software Quality-Related Issues or Concerns 4-8
4.3.4 Recommendations 4-9
4.4 TOPICAL AREA 4 ASSESSMENT: DESIGN PHASE 4-9
4.4.1 Criterion Specification and Result 4-9

4.4.2 Sources and Method of Review *4-12*
4.4.3 Software Quality-Related Issues or Concerns *4-13*
4.4.4 Recommendations *4-13*
4.5 TOPICAL AREA 5 ASSESSMENT: IMPLEMENTATION PHASE 4-13
4.5.1 Criterion Specification and Result *4-13*
4.5.2 Sources and Method of Review *4-14*
4.5.3 Software Quality-Related Issues or Concerns *4-14*
4.5.4 Recommendations *4-14*
4.6 TOPICAL AREA 6 ASSESSMENT: TESTING PHASE 4-14
4.6.1 Criterion Specification and Result *4-14*
4.6.2 Sources and Method of Review *4-16*
4.6.3 Software Quality-Related Issues or Concerns *4-16*
4.6.4 Recommendations *4-16*
4.7 TOPICAL AREA 7 ASSESSMENT: USER INSTRUCTIONS 4-16
4.7.1 Criterion Specification and Result *4-17*
4.7.2 Sources and Method of Review *4-17*
4.7.3 Software Quality-Related Issues or Concerns *4-17*
4.7.4 Recommendations *4-18*
4.8 TOPICAL AREA 8 ASSESSMENT: ACCEPTANCE TEST 4-18
4.8.1 Criterion Specification and Result *4-18*
4.8.2 Sources and Method of Review *4-19*
4.8.3 Software Quality-Related Issues or Concerns *4-19*
4.8.4 Recommendations *4-19*
4.9 TOPICAL AREA 9 ASSESSMENT: CONFIGURATION CONTROL 4-19
4.9.1 Criterion Specification and Result *4-19*
4.9.2 Sources and Method of Review *4-20*
4.9.3 Software Quality-Related Issues or Concerns *4-20*
4.9.4 Recommendations *4-20*
4.10 TOPICAL AREA 10 ASSESSMENT: ERROR IMPACT 4-20
4.10.1 Criterion Specification and Result *4-20*
4.10.2 Sources and Method of Review *4-22*
4.10.3 Software Quality-Related Issues or Concerns *4-22*
4.10.4 Recommendations *4-22*
4.11 TRAINING PROGRAM ASSESSMENT 4-23
4.12 SOFTWARE IMPROVEMENTS AND NEW BASELINE 4-23

5.0 Conclusions **5-1**

6.0 Acronyms and Definitions **6-1**

7.0 References **7-5**

APPENDIX A. SOFTWARE INFORMATION TEMPLATE A-1

TABLES

Page

Table 1-1 — Plan for SQA Evaluation of Existing Safety Analysis Software 1-4

Table 1-2 — Summary Description of the MELCOR Software in the Context of LPF Analysis 1-7

Table 1-3 — Software Documentation Reviewed for MELCOR (LPF Applications) 1-10

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

Table 2-2 — Summary of Important Recommendations for MELCOR for LPF Applications 2-2

Table 3-1 — Lessons Learned 3-3

Table 4.0-1 — Cross-Reference of Requirements with Subsection and Entry from DOE (2003e) 4-1

Table 4.1-1 — Subset of Criteria for Software Classification Topic and Results 4-3

Table 4.2-1 — Subset of Criteria for SQA Procedures and Plans Topic and Results 4-6

Table 4.3-1 — Subset of Criteria for Requirements Phase Topic and Results 4-8

Table 4.4-1 — Subset of Criteria for Design Phase Topic and Results 4-9

Table 4.5-1 — Subset of Criteria for Implementation Phase Topic and Results 4-13

Table 4.6-1 — Subset of Criteria for Testing Phase Topic and Results 4-14

Table 4.7-1 — Subset of Criteria for User Instructions Topic and Results 4-17

Table 4.8-1 — Subset of Criteria for Acceptance Test Topic and Results 4-18

Table 4.9-1 — Subset of Criteria for Configuration Control Topic and Results 4-19

Table 4.10-1 — Subset of Criteria for Error Impact Topic and Results 4-21

Table 4.12-1 — Comparison of SQA Upgrade Steps Discussed in Bixler (2000) with the Approach Discussed in DOE (2003e) 4-24

INTENTIONALLY BLANK

FIGURES

Page
Figure 1-1 MELCOR Execution Flowchart.....1-6

INTENTIONALLY BLANK

Software Quality Assurance Improvement Plan: MELCOR 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 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 Methods for Estimation of Leakages and Consequences of Releases (MELCOR) software is one of the codes designated for the toolbox. It is being evaluated for leak path factor (LPF) applications. To determine the actions needed to bring the MELCOR code into compliance with the SQA qualification criteria in the context of LPF applications 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 MELCOR against identified criteria.

The balance of this document provides the outcome of the 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 MELCOR 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 MELCOR for supporting safety analysis. Informed use of the software can be assisted by the current set of MELCOR reports (refer to Table 1-3), and the code guidance report for DOE safety analysts, *MELCOR Computer Code Application Guidance for Leak Path Factor in Documented Safety Analysis*, (DOE, 2004). Furthermore, while SQA improvement actions are recommended for MELCOR, no evidence has been found of programming, logic, or other types of software errors in MELCOR 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"), five requirements are met at acceptable level, i.e., *Software Classification, Implementation Phase, User Instructions*, *Acceptance Test,* and *Configuration Control*; Requirements 1, 5, 7, 8, and 9 respectively. Improvement actions are recommended to meet SQA criteria for the remaining five requirements. This evaluation outcome is deemed acceptable because: (1) MELCOR 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 MELCOR is limited to those analytic applications for which the software is intended.

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

1. Correcting known defects in the SQA process
2. Upgrading existing SQA documentation
3. Providing training on a regular basis, and
4. Revising and developing new software documentation.

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

- Updated Software Quality Assurance Plan
- Software Requirements Document (Specific to LPF)
- Software Design Document (Specific to LPF)
- Test Case Description and Report (Specific to LPF)
- Updated Software Configuration and Control
- Updated Error Notification and Corrective Action Report Procedure, and
- Updated User's Manual.

Approximately two full-time equivalent years is conservatively estimated to upgrade MELCOR 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 MELCOR is planned for release in the future. It is recommended that this version be evaluated upon issue relative to the software improvement and baseline 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 1.8.5 as the designated version of the software for the toolbox.

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

1.0 Introduction

This document reports the results of a gap analysis for Version 1.8.5 of the MELCOR computer code in the context of LPF applications. The intent of the gap analysis is to determine the actions needed to bring the specific software into compliance with established SQA criteria. A secondary aspect of this report is to develop an estimate of the level of effort required to upgrade each code based on the gap analysis results.

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

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

While progress was made in resolving several of the issues raised in TECH-25, the DNFSB issued Recommendation 2002-1 on *Quality Assurance for Safety-Related Software* in September 2002. The DNFSB enumerated many of the points noted earlier in TECH-25, but noted specific concerns regarding the quality of the software used to analyze and guide safety-related decisions, the quality of the software used to design or develop safety-related controls, and the proficiency of personnel using the software. The Recommendation identified a number of quality assurance issues for software used in the DOE facilities for analyzing hazards, and designing and operating controls that prevent or mitigate potential accidents. The development and maintenance of a collection, or "toolbox," of high-use, SQA-compliant safety analysis codes is one of the major commitments contained in the 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, MELCOR Version 1.8.5 will likely require some degree of quality assurance improvement before meeting current SQA standards. The analysis documented herein is an evaluation of MELCOR, in the context of LPF applications, relative to current software quality assurance criteria. It assesses the extent of the deficiencies, or gaps, to provide DOE and the software developer the extent to which minimum upgrades are needed. The overall assessment is therefore termed a "gap" analysis.

1.2 Evaluation of Toolbox Codes

The quality assurance criteria identified in later sections of this report are defined as the set of established requirements, or 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 analysis 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 MELCOR, 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 MELCOR development program. While independent of the code developer, the SQA evaluators responsible for MELCOR 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

This gap analysis is applicable to the MELCOR code, one of the six designated toolbox codes for safety analysis, for applications of LPF analysis. While the MELCOR code is the subject of the current report,

other safety analysis software considered for the toolbox in the future may be evaluated with the same process applied here. The template outlined in this document is applicable for any analytical software as long as the primary criteria are ASME NQA-1, 10 CFR 830, and related DOE directives discussed in DOE (2003e).

1.5 Purpose

The purpose of this report is to document the gap analysis performed on the MELCOR code for LPF applications as part of DOE's implementation plan on SQA improvements.

1.6 Methodology for Gap Analysis

The gap analysis for MELCOR (LPF applications) 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 SQA to assess the quality of the MELCOR code in the context of LPF applications. 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.

An information template was transmitted to the Safety Analysis Software Developers on 20 October 2003 to provide basic information as input to the gap analysis process. The main section of the template is attached as Appendix A to the present report, with an example section and references removed. No written response to the information template has been provided by the MELCOR software developers. Instead, SNL personnel were interviewed in January 2004 to obtain needed information to perform this analysis. The information in Appendix A was used as a guide during this interview, and the results are captured in the details of this report, Section 4.0.

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

Phase	Procedure
1. Prerequisites	a. Determine that sufficient information is provided by the software developer to allow it to be properly classified for its intended end-use. b. Review SQAP per applicable requirements in Table 3-3 of DOE (2003e).
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.
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: • SQA 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.

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

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

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

1.7 Summary Description of Software Being Reviewed

The gap analysis was performed on Version 1.8.5 of the MELCOR code in the context of LPF applications. MELCOR (Gauntt, 2000a) is a generalized mass transport and thermal hydraulic computer program. MELCOR is available for the UNIX workstation platform as well as the PC platform. MELCOR 1.8.5 is the latest released version of MELCOR as of the beginning of this assessment. A patch was released 10/23/2001 (see the SNL MELCOR site http://melcor.sandia.gov/.) MELCOR Version 1.8.5 includes NRC and DOE sponsored changes made over the years.

MELCOR is a fully integrated, engineering-level computer code whose primary purpose is to model the progression of accidents in light water reactor nuclear power plants. A broad spectrum of severe accident phenomena in both boiling and pressurized water reactors is treated in MELCOR in a unified framework. MELCOR estimates fission product source terms and their sensitivities and uncertainties in a variety of applications. The MELCOR code is composed of a number of major modules, or packages, that together model the major systems of a reactor plant and its generally coupled interactions.

MELCOR was initially developed at the Sandia National Laboratories (SNL) under the sponsorship of the USNRC to assess reactor severe accident conditions. MELCOR was developed as a "research" code by the NRC and SNL. It was intended to be used to perform parametric studies, scoping studies, and studies to check the results of other models. For the last several years, MELCOR has been used in the DOE Complex to model release of radioactive airborne material from nuclear facilities and structures. The amount released is termed leakage and is usually expressed as a fraction of the amount considered available for release. This fraction released is referred to as the Leak Path Factor, LPF.

Although the MELCOR computer code was developed to model the progression of accidents in light water reactor nuclear power plants, the modeling capabilities of MELCOR are sufficiently flexible that it can be applied to the analysis of nonreactor problems. When performing LPF studies for nuclear facilities the modules used are reduced (through input specification) to those which will enable the modeling of the release and transport of aerosolized materials – the code activates modules based on the input card identification field. The most common modules used for Leak Path Factor analyses are:

- Executive Package (EXEC)
- Non-Condensable Gas Package (NCG)
- Control Volume Hydrodynamics Package (CVH)
- Flow Path Package (FL)
- Heat Structures Package (HS)
- Radio-Nuclide Package (RN)

- Control Function Package (CF)
- Tabular Function Package (TF)

Both NRC and the DOE have sponsored changes to the code, with NRC being the primary sponsor. For example, modifications were made to a version of MELCOR to model K reactor severe accidents at the DOE operated Savannah River Site. Some of this work factored into later updates of the code.

Figure 1-1 depicts a basic flowchart showing the steps required to successfully execute MELCOR.

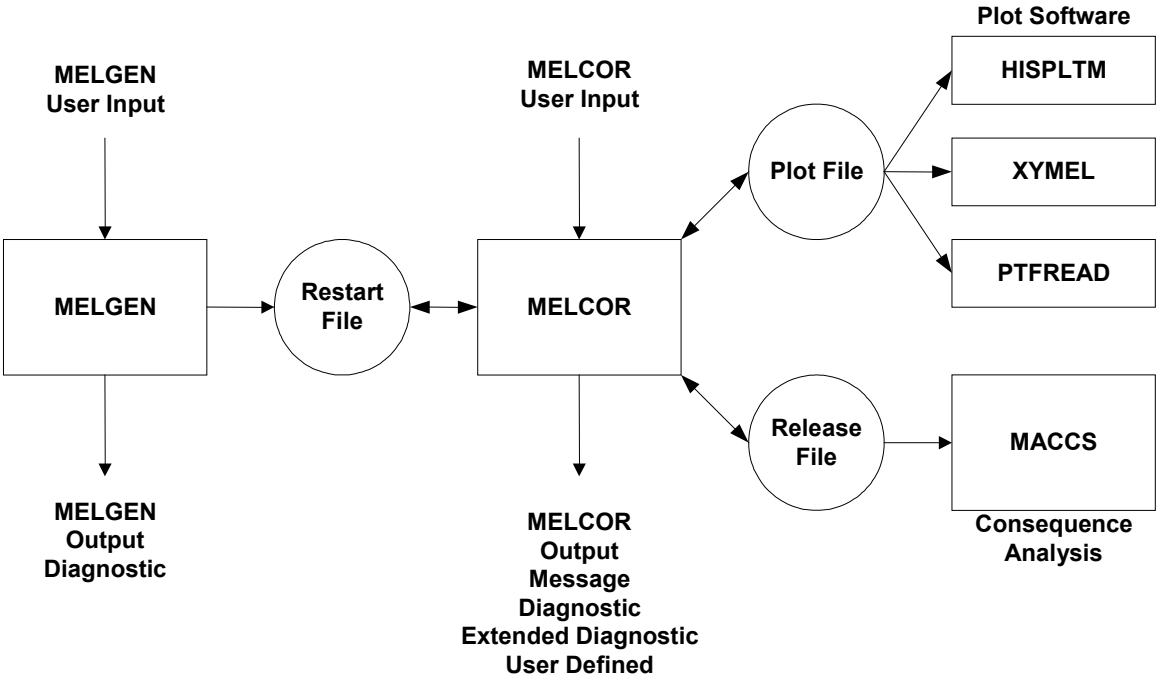

Figure 1-1 MELCOR Execution Flowchart

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

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

Table 1-2 — Summary Description of the MELCOR Software in the Context of LPF Analysis

Type	**Specific Information**
Code Name	MELCOR - Methods for Estimation of Leakages and Consequences of Releases
Developing Organization and Sponsor	Sandia National Laboratories (SNL) for the U.S. Nuclear Regulatory Commission (primary), International Cooperative Severe Accident Research Program (CSARP) and U.S. Department of Energy (minor contribution)
Version of the Code	Version 1.8.5
Auxiliary Codes	AUXILIARY CODES: The plotting software distributed with MELCOR includes HISPLTM, XYMEL, and PTFREAD. The output from MELCOR can be input into the MACCS2 (or earlier version MACCS) code to perform consequence analysis. MELCOR INSTALL Installs software.
Software Platform/Portability	FORTRAN 77/90, PC based some system dependencies. Also runs on Unix (not tested for every platform), source code is available for HP, SUN and others.
Coding and Computer	Fortran 77, PC based 80486 or Pentium processor (C00652/PC486/00).
Technical Support	R. O. Gauntt Sandia National Laboratories P.O. Box 5800 Albuquerque, NM 87185-0748 (505) 284-3989 rogaunt@sandia.gov;
Code Procurement	The MELCOR program and comprehensive set of MELCOR documentation is available through SNL. MELCOR has a website: http://melcor.sandia.gov/. Permission from NRC is needed to acquire the code.
Code Package	Included are the references cited below. Also included are the Fortran source code, and an executable file. Training slides and a sample input deck are also available on the web site.

Table 1-2 — Summary Description of MELCOR Software in the Context of LPF Analysis (Continued)

Documentation Supplied with Code Transmittal	1. Gauntt, 2000a, Gauntt et al., *MELCOR Computer Code Manuals, Vol. 1: Primer and Users' Guide*, Version 1.8.5, NUREG/CR-6119 Rev. 2, SAND2000-2417/1, May 2000. 2. Gauntt, 2000b, Gauntt et al., *MELCOR Computer Code Manuals, Vol. 2: Reference Manuals*, Version 1.8.5, NUREG/CR-6119 Rev. 2, SAND2000-2417/2, May 2000. 3. Gauntt, 2001, Gauntt et al., *MELCOR Computer Code Manuals, Vol. 3: Demonstration Problems*, Version 1.8.5, NUREG/CR-6119 Rev. 0, SAND2001-0929P, May 2001. (Available upon request) 4. File of electronic input decks. 5. MELCOR INSTALLER. 6. Instructions for installing MELCOR for use with Digital Fortran 5/6 and Developer Studio.
Nature of Problem	MELCOR is a fully integrated, relatively fast-running code that models the progression of severe accidents in nuclear power plants. An entire spectrum of severe accident phenomena is modeled in MELCOR. Characteristics of severe accident progression that can be treated with MELCOR include the thermal-hydraulic response in the reactor coolant system, reactor cavity, containment, and confinement buildings; core heatup and degradation; radionuclide release and transport; hydrogen production, transport, and combustion; core-concrete attack; heat structure response; and the impact of engineering safety features on thermal-hydraulic and radionuclide behavior. For applications in non-reactor facilities of the DOE Complex, MELCOR has been used primarily to model in-facility transport of the release of radioactive airborne material. Deposition inside the building is calculated and the leakage to the outside environment is expressed as a fraction of the amount considered available for release and is termed the LPF.
Method of Solution	MELCOR can be used to model in-facility transport that involves the two broad areas of mixing/transport of a hazardous gas and/or aerosol transport of a hazardous material. MELCOR employs the control volume approach with lumped parameter models. MELCOR has detailed mechanistic aerosol dynamics models for the transport, deposition, and agglomeration of aerosols. Major assumptions in MELCOR include: • Each control volume gas space is well mixed, except each cell does allow for a pool covered by a gas volume. • Each gas species has the same velocity in the flow path connections. • Noncondensable gases are assumed to be ideal. • Turbulence and species diffusion within a control volume are not modeled, except in the aerosol model and condensation/evaporation on surfaces.

Table **1-2 — Summary Description of MELCOR Software in the Context of LPF Analysis (Continued)**

Restrictions or Limitations	The control-volume, lumped-parameter approach of MELCOR does not model multi-dimensional effects, such as stratification of gases within a room. (To overcome this, one approach is to break the room into more volumes sometimes coupling the approach with computational fluid dynamics (CFD) code results.)
Run Time	The typical execution time depends on machine, detail of the model, and the length of the transient. Runtimes on the CRAY vary from 0.1 s to on the order of 1 h.[2] Runtimes for the Marviken-V Aerosol Transport Tests ATT varied from 3442 cpu(s) on a CRAY XMP-24, to 26,700 cpu(s) on a SUN Sparc2. Detailed code calculation of 24-h LaSalle Station Blackout calculation was 2 h on an HP. Simplified code calculation runtime for a 4-h sample problem transient was 15 min on an HP. The ratio of real time to runtime can vary from 0.5 to 100, depending on the nodalization.
Computer Hardware Requirements	Memory requirement is 5 MB. Depending on the model application Gigabytes of storage for output files may be required.[2]
Computer Software Requirements	MELCOR is available for the UNIX workstation platform as well as the PC platform. The execution of MELCOR on a PC is very efficient and user friendly. While either platform may be used, simply because of ease of use the latter is recommended. (A benefit of running on a PC is the ease with which output data can be processed in spreadsheet or text file programs.)
Other Versions Available	No other versions are available from SNL. INEEL and SRS both have developed specialized versions, but these are not supported by SNL and the sponsors.

[2] The data in this paragraph is dated by about 10 years. Typical run times on today's computers would be a few minutes. The most complicated models run approximately one week. Storage (output file size) is often more of limit today than run time. Actual conditions will depend on the hardware and the type of problem being executed.

Table 1-3 — Software Documentation Reviewed for MELCOR (LPF Applications)

No.	Reference
1.	Gauntt, 2000a, Gauntt et al., *MELCOR Computer Code Manuals, Vol. 1: Primer and Users' Guide*, Version 1.8.5, NUREG/CR-6119 Rev. 2, SAND2000-2417/1, May 2000.
2.	Gauntt, 2000b, Gauntt et al., *MELCOR Computer Code Manuals, Vol. 2: Reference Manuals*, Version 1.8.5, NUREG/CR-6119 Rev. 2, SAND2000-2417/2, May 2000.
3.	Gauntt, 2001, Gauntt et al., *MELCOR Computer Code Manuals, Vol. 3: Demonstration Problems*, Version 1.8.5, NUREG/CR-6119 Rev. 0, SAND2001-0929P, May 2001.
4.	SNL, 2001, Sandia National Laboratories. *5th MELCOR User's Workshop*, Bethesda, MD, May 10th – 15th, 2001.
5.	SNL 2003, Sandia National Laboratories. Nuclear Waste Management Procedure, NP 19-1, *Software Requirements*, Revision 10, Waste Isolation Pilot Plant, (May 2003).
6.	East, 1998, J.M. East and E.P. Hope, *Independent Evaluation of the MACCS2 Software Quality Assurance Program (U)*, WSRC-RP-98-00712, Westinghouse Savannah River Company, Aiken, SC (August 1998).
7.	DNFSB, 2000, Defense Nuclear Facilities Safety Board, *Quality Assurance for Safety-Related Software at Department of Energy Defense Nuclear Facilities*, Technical Report DNFSB/TECH-25, (January 2000).
8.	DOE 2004, U.S. Department of Energy. *MELCOR Computer Code Application Guidance for Leak Path Factor in Documented Safety Analysis*, (May 2004). Updates to this report are available at the DOE/EH Central Registry: http://www.eh.doe.gov/sqa/central_%20registry/MELCOR/melcor.htm
9.	SNL 1992, Sandia National Laboratories. *Software Quality Assurance Procedures for MELCOR*, Revision 1.2, (August 1992).

2.0 Assessment Summary Results

2.1 Criteria Met

Of the ten general topical quality areas assessed in the gap analysis, five satisfactorily met the criteria. The analysis found that the MELCOR SQA program (in the context of LPF applications) in general, met criteria for *Software Classification, Implementation Phase, User Instructions*, *Acceptance Test,* and *Configuration Control*, Requirements 1, 5, 7, 8, and 9 respectively. Five topical quality areas were not met satisfactorily. The major deficiency areas are covered below in Section 2.2 (Exceptions to Requirements). Detail on the evaluation process relative to the requirements and the criteria applied are found in Section 4.

2.2 Exceptions to Requirements

Some of the more important exceptions to criteria found for MELCOR 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)	SQA Plan and Procedures for Version 1.8.5 of MELCOR software were lacking components to match present day requirements. Portions of the existing version are out of date or are not currently followed.	As part of the new software baseline, the SQA Plan covering version 1.8.5 and successor versions of MELCOR should be provided to the Central Registry. SQA procedures that provide prescriptive guidance to the MELCOR software developers should be made available to a SQA evaluator for confirmatory review.
			Establish a written and approved SQA plan eliminating draft or non-compliant informal processes of development.
			Upgrade SQA program documentation, especially those procedures used for new features added in MELCOR that have an effect on modules that are typically used in LPF applications. Ensure prompt defect/error reporting.
2.	Requirements Phase (Section 4.3)	A Software Requirements Document for Version 1.8.5 of MELCOR is not available.	As part of the new software baseline for MELCOR, a Software Requirements Document should be prepared.

No.	Criterion	Reason Not Met	Remedial Action(s)
3.	Design Phase (Section 4.4)	A Software Design Document is not available. Thus, design information was not directly available. Instead, it was necessary to infer the intent of MELCOR design from model description and user guidance documents.	As part of the new software baseline for MELCOR, a Software Design Document should be prepared.
4.	Testing Phase (Section 4.6)	A Software Testing Report Document has not been produced for MELCOR, and therefore, test process and methodology could not be evaluated directly. Thus, testing process and methods had to be inferred from other information. Isolated validation studies have been previously documented for various phenomenological areas, including aerosol transport, which is the key area for LPF applications. While these studies promote confidence in the models for LPF applications, the necessary formality is lacking to make a complete evaluation.	As part of the new software baseline for MELCOR, a test case report should be prepared. An important part of the new baseline set of documentation should specifically address aerosol transport phenomena and LPF applications.
5.	Error Notification (Section 4.10)	An Error Notification and Corrective Action Report process is in place at SNL, but limited documentation is available. Users are not necessarily notified of errors. Follow up with the notifying agent is not always guaranteed, and the impact is not always assessed and reported.	While a Software Problem Reporting system is in place at SNL, it requires revision to ensure affected users are notified, closure occurs with the originator, and impact determinations are completed promptly.

2.3 Areas Needing Improvement

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

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

Table 2-2 — Summary of Important Recommendations for MELCOR for LPF Applications

No.	UI – User Interface Enhancements TM – Technical Model Upgrade	Recommendation
1.	UI	Expand selection of sample problems to include those problems and releases type that are often treated in LPF analysis for Documented Safety Analyses (DSAs).
2.	UI	Provide the user more control over the printed output by allowing only selected items to print. This will help avoid

No.	UI – User Interface Enhancements TM – Technical Model Upgrade	Recommendation
		lengthy output files, and enhance post-processing. As an example, similar print options as used in MACCS2 would be useful. Consider adding in this same update an option to print summary information on the aerosol mass balance amongst volumes. This would consolidate information currently available that the user must manually extract at present, and would lessen the likelihood of error.

Item 1 in the above table will serve at least two functions. First, it will serve to enhance training for LPF. Second, it will support the LPF testing and SQA changes identified in other areas of this report.

2.4 Conclusion Regarding Software's Ability to Meet Intended Function

The MELCOR 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, MELCOR *Computer Code Application Guidance for Leak Path Factor in 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 MELCOR 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.

3.0 Lessons Learned

Table 3-1 provides a summary of the lessons learned during the performance of the MELCOR 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 MELCOR, but independent confirmation of the SQA program, practices, and procedures is not possible due to lack of documentation.
2.	Observance of SQA requirements in the development of safety analysis software has not been consistent. It appears to be sporadic in application, poorly funded, and performed as an add-on activity. (Note that this is consistent with the "research" specification as given to the code.) Funding level during program development has been a key factor in determining the level of attention to SQA and the adequacy of documentation.
3.	While some evidence of pre-development planning is found for the MELCOR 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. Initial rough estimates are 2 full-time equivalent years and should be a high priority. As time passes, knowledgeable personnel may become unavailable and it will become more difficult and costly (if not impossible) to document the QA status of the code.
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 MELCOR LPF applications and other designated software for the DOE toolbox.

4.0 Detailed Results of the Assessment Process

Ten topical areas, or requirements, are presented in the assessment as listed in Table 4.0-1. Training and Software Improvements sections follow the 10 topical areas. Included in the software improvements section is an estimate of the resources required to upgrade MELCOR.

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 the software on the MELCOR website (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 MELCOR software for LPF calculations in safety analysis applications would be expected to interpret the information on the software in light of the requirements that are discussed in Appendix A to DOE-STD-3009-94 to decide on an appropriate safety classification.

For most organizations, the safety class or safety significant classification, or Level B in the classification hierarchy discussed in DOE (2003e), would be selected. In the software requirements procedure provided by SNL, the MELCOR 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 by the MELCOR users' manuals that are available from the software developer and the MELCOR website. Interpreted in light of Appendix A to DOE-STD-3009-94.

4.1.2 Sources and Method of Review

Documentation supplied with the MELCOR software package.

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).

Use is made of an earlier independent review of the MACCS2 SQA Program (East 1998) coupled with an interview of the Sandia National Laboratories authors to determine the level of compliance with this requirement.

While the (East 1998) review focused on the MACCS2 computer code, much information was obtained on the general SQA program that existed at SNL around the time that both MACCS2 and the MELCOR software were being developed. The documented review was preceded by an in-depth review at Sandia National Laboratories in 1997. The following, based on the earlier review, provides a good synopsis of the SQA program that existed in the late 1980s and early 1990s.

SNL established a SQA program for Laboratory software in the late 1980s and early 1990s that was compliant with the IEEE Standard for SQA Plans. The final volume was put into place in 1995. The guidelines[3] are documented as shown:

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

Volume 1 – Software Quality Planning [SNL, 1987]
Volume 2 – Documentation [SNL, 1995]
Volume 3 – Standards, Practices, and Conventions [SNL, 1986]
Volume 4 – Configuration Management [SNL, 1992a]; and
Volume 5 –Tools, Techniques, and Methodologies [SNL, 1989].

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

During the interview conducted with SNL personnel in January 2004, the MELCOR SQA procedures document (SNL-1992b) was provided and reviewed. The SNL(1992b) document provides SQA plan detailed information specific to MELCOR. It references (SNL 1986, SNL 1987, and SNL 1989) discussed above as primary documents. Topics covered include:

- Maintenance Procedures
  - Configuration Identification
  - Alternate Software Packages
- The Defect Investigation Report (DIR) Process
  - Request Description
  - Diagnosis
  - Resolution Plan
  - Change/Testing
  - Update Implementation
- Documenting Actions Not Involving Code Changes
- Configuration Status Accounting
- Validation and Verification of MELCOR
- MELCOR User's Guides and Reference Manuals
- Testing and Review for Code Release
- Tools, Techniques and Methodologies
- Code Written by External Suppliers
- Special Purpose Code Modifications

This plan was followed during the 1990's as MELCOR was developed and modified. The authors continue to follow the plan today, with less rigidity and with some modification as funding allows.

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. Based on the SQA Program review from 1997-1998 (J. East), and East (1998), it can be inferred from the general SNL SQA information and MACCS2-specific details that most elements of a compliant SQA plan and procedures were likely in place and followed during the development of MELCOR version 1.8.5. This was confirmed by meetings with the code authors in January 2004. However, definitive confirmation through written, approved documentation is not always 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.	Yes.	(SNL 1992b) outlines the MELCOR software assurance plan and the procedures in place when MELCOR was developed.
2.2	Verify that procedures/plans for SQA (**SQA Plan**) have identified software engineering methods.	Yes.	(SNL 1992b) provides coding guidelines as well as steps for modifying or adding code.
2.3	Verify that procedures/plans for SQA (**SQA Plan**) have identified documentation to be required as part of program.	Yes.	(SNL 1992b) Section 4.0 provides direct reference to and plans for user's guides and reference manuals
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.	Yes.	(SNL 1992b) provides standards for coding, techniques for modifying the coding and methods to be used in program development.
2.5	Verify that procedures/plans for SQA (**SQA Plan**) have identified software reviews and schedule.	Partial.	Elements of this existed based on discussions with the authors. Software reviews were conducted. Schedules for the reviews and evidence for the thoroughness of the reviews were not found in the available documentation. (SNL 1992b) discusses testing and review in Section 5.0.
2.6	Verify that procedures/plans for SQA (**SQA Plan**) have identified methods for error reporting and corrective actions.	Yes. (Recently less rigor)	(SNL-1992b) provides discussion of the DIR (Defect Investigation Report) process. Discussion with SNL in January 2004 indicates the DIR process was rigorously followed during the 90's. With decreasing funding, error reporting has continued, but is less rigorous, with corrective actions requiring more time. Documentation and notification is less rigorous.

4.2.2 Sources and Method of Review

This review was based initially on the general SNL SQA information and the MACCS2-specific information from East (1998) and making inferences to the MELCOR code that was developed around the same timeframe as MACCS2 (MELCOR 1.8.0 released in March of 1989 and the current version 1.8.5 was released October 2000; development of MACCS2 began in 1992 with the release of the current

version 1.12 occurring in 1997). This was later supported by meetings with SNL in January 2004 specifically to discuss SQA for MELCOR. The primary reference for the SQA plan was provided in this meeting as (SNL-1992b). This plan refers to the same governing SQA documents as used by MACCS2 and reported on by East.

4.2.3 Software Quality-Related Issues or Concerns

An SQA plan for MELCOR exists. The plan is dated and consideration should be given to revising it to conform to current practices being followed for MELCOR and current day SQA expectations.

The SQA plan lacks guidance for providing design requirements for modifications being made for the code.

The SQA plan lacks detailed guidance on testing of newly developed software or modifications. Guidance should concentrate on level of testing required, type of testing, and independent verification of coding. Documentation requirements for code testing appear to be lacking. Currently modifications are made and tested against experimental results. In fact, most recent modifications are planned specifically to match to a particular type of result or experiment. This gives a level of confidence in the overall results. Testing of the coding on a line-by-line basis and for quality was not evident in the available documentation for the SQA plan although it is known this was done with varying degrees of rigor during development.

The SQA plan should address prompt error and impact notification to users. Currently (SNL-1992b) requires users be notified if funding is available. Errors or deficiencies are usually reported via email. These are then logged and if code modifications are made, they are incorporated into a future version of the code. Recently no major errors have been discovered. It may take many months for modifications resulting from any given email to be incorporated into the code and released. Not all users are notified of code modifications being made due to these emails. Documentation of detailed closure with the original email author is lacking or not formalized.

4.2.4 Recommendations

Recommendations related to this topical area are provided as follows:

- Develop an updated SQA plan for Version 1.8.5 of MELCOR (at least as the code relates to LPF analysis). (Revise as needed for future updates released for public distribution).
    - Ensure the update is consistent with the current technology and practices.
    - Ensure the plan provides specific guidance regarding design requirements and documentation of design requirements.
    - Ensure the plan addresses prompt defect/error notification to users. (At least as the errors relate to LPF analyses)

4.3 Topical Area 3 Assessment: Requirements Phase

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

4.3.1 Criterion Specification and 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.	Partial	A verifiable, written set of software requirements is lacking. Requirements for modifications are given verbally/contractually with NRC.
3.2	Software requirements are specified, documented, reviewed and approved.	Partial.	In earlier MELCOR development efforts, written hypothetical coding plans were generated. In practice, this was found not to be beneficial and the plans would be completely rewritten or pitched. Current modifications do not generate comparable initial guidance. A verifiable, written set of software requirements is lacking.
3.3	Requirements define the functions to be performed by the software and provide detail and information necessary to design the software.	Partial.	A verifiable, written set of software requirements is lacking.
3.4	A **Software Requirements Document**, or equivalent defines requirements for functionality, performance, design inputs, design constraints, installation considerations, operating systems (if applicable), and external interfaces necessary to design the software.	Partial.	A verifiable, written set of software requirements is lacking. The contractual agreements for code development with NRC do lay out top-level direction year to year.
3.5	Acceptance criteria are established in the software requirements documentation for each of the identified requirements.	No.	A verifiable, written set of software requirements is lacking. Judgment is used as modeling progresses to discern the adequacy of model changes, usually against experiments.

4.3.2 Sources and Method of Review

This review was based on based on discussion with SNL in January 2004 and information contained in East (1998), Gauntt (2000a), Gauntt (2000b), Gauntt (2001), and (SNL 1992b).

4.3.3 Software Quality-Related Issues or Concerns

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

4.3.4 Recommendations

Develop a Software Requirements Document for MELCOR. At a minimum, this document should address requirements related to LPF applications for meeting the prerequisites for the DOE toolbox. A broader approach would consider NRC-specified needs for the software as well and address the full capabilities of the code.

4.4 Topical Area 4 Assessment: Design Phase

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

A Software Design Document has not been provided by the MELCOR software developers. To permit a limited evaluation, an alternative process was employed of reviewing MELCOR documentation for evidence that criterion requirements were met at least partially in an informal manner.

4.4.1 Criterion Specification and Result

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

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
4.1	The software design was developed, documented, reviewed and controlled.	Partial.	Elements of this criterion may be inferred from code user documentation, reference manuals and discussions with SNL.
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.	(SNL 1992b) provides significant detail in some area on code design and modeling constraints. Similar constraints were understood by the developers when not documented on paper. Documented design requirements were lacking, therefore, documentation of having met requirements is lacking.
4.3	The following design should be present and documented: the design should specify the interfaces, overall structure (control and data flow) and the reduction of the overall structure into physical solutions (algorithms, equations, control logic, and data structures).	Yes.	Inferred from MELCOR documentation.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
4.4	The following design should be present and documented: that computer programs were designed as an integral part of an overall system. Therefore, evidence should be present that the software design considered the computer program's operating environment.	Yes.	Inferred from MELCOR documentation.
4.5	The following design should be present and documented: evidence of measures to mitigate the consequences of software design problems. These potential problems include external and internal abnormal conditions and events that can affect the computer program.	Partial.	The documentation of a systematic effort in this area is lacking. Practical steps were taken by the code developers to handle abnormal conditions. For example, the code developers do not let the code stop execution without a message log. Bugs and problems have been corrected over the years when found.
4.6	A Software Design Document, or equivalent, is available and contains a description of the major components of the software design as they relate to the software requirements.	No.	While there is some evidence of the design relating back to requirements as set out for the code contractually with the sponsor, there was no formal documentation available and little evidence of a systematic effort to tie final design to a set of initial requirements.
4.7	A Software Design Document, or equivalent, is available and contains a technical description of the software with respect to the theoretical basis, mathematical model, control flow, data flow, control logic, data structure, numerical methods, physical models, process flow, process structures, and applicable relationship between data structure and process standards.	Partial.	A set of the listed elements is addressed in documentation (see Section 4.4.2 of this report). Most of the models, etc. are described in detail. A formal design document was not initially generated as a part of each modification process. The authors would informally sketch out the modifications to be made. Final models as developed would normally be incorporated in the User's Manual or Reference Manuals, for major changes.
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	Formal design documents are lacking. However, with the supplied documentation and some experience it is possible to understand if inputs/outputs are logical and within range.
4.9	A Software Design Document, or	Yes.	Formal design documents are

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	equivalent, is available and contains the design described in a manner that can be translated into code.		lacking. However, with the supplied documentation and some experience, it is possible to translate the models and theories as described to code.
4.10	A Software Design Document, or equivalent, is available and contains a description of the approach to be taken for intended test activities based on the requirements and design that specify the hardware and software configuration to be used during test execution.	Partial.	Documentation is lacking. Most modifications are initiated as part of a project to compare to test data or experiment.
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.	Evidence of substantial peer review exists. Documentation of completeness is difficult to corroborate. Documentation of pre-planning in software design documents is lacking.
4.12	The organization responsible for the design assured that the test results adequately demonstrated the requirements were met.	Partial.	A verifiable, written set of documentation of software design requirements is lacking. Evidence exists that substantial testing was performed.
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.	Significant independent review has been performed. Documentation of reviewer qualifications and independence is lacking. For example, there is evidence of peer review during the 1990-91 timeframe from training slide material that is available from the MELCOR website (SNL, 2001). The NRC reviews code modules when completed by SNL.
4.14	The results of the Independent Review are documented with the identification of the verifier indicated.	Partial.	Significant independent review has been performed. Complete documentation is lacking.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
4.15	If review alone was not adequate to determine if requirements are met, alternate calculations were used, or tests were developed and integrated into the appropriate activities of the software development cycle.	Partial.	A verifiable, written set of documentation of software design requirements is lacking. Significant independent review has been performed. The code has been modified over the years and tested to provide reasonable assurance the models are adequate.
4.16	Software design documentation was completed prior to finalizing the Independent Review.	Partial.	Some review was known to have been conducted in parallel with design documentation preparation or before preparation of its equivalent.
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.	Partial.	Integrated documentation of the design requirements is lacking, as is documentation of the review detail and its bases. Judgment was used by the code developers to determine what would be reviewed and when. MELCOR has undergone many man-years of independent review and is believed to be robust. Elements of this activity have been documented by various organizations at various times for varying applications and models.

4.4.2 Sources and Method of Review

SNL personnel were interviewed in January 2004. Design requirements were evaluated through review of the following documents:

Gauntt, 2000a, Gauntt et al., *MELCOR Computer Code Manuals, Vol. 1: Primer and Users' Guide*, Version 1.8.5, NUREG/CR-6119 Rev. 2, SAND2000-2417/1, May 2000.

Gauntt, 2000b, Gauntt et al., *MELCOR Computer Code Manuals, Vol. 2: Reference Manuals*, Version 1.8.5, NUREG/CR-6119 Rev. 2, SAND2000-2417/2, May 2000.

Gauntt, 2001, Gauntt et al., *MELCOR Computer Code Manuals, Vol. 3: Demonstration Problems*, Version 1.8.5, NUREG/CR-6119 Rev. 0, SAND2001-0929P, May 2001.

SNL, 2001, Sandia National Laboratories. *5th MELCOR User's Workshop*, Bethesda, MD, May 10th – 15th, 2001.

SNL 2003, Sandia National Laboratories. Nuclear Waste Management Procedure, NP 19-1, *Software Requirements*, Revision 10, Waste Isolation Pilot Plant, (May 2003).

SNL (1992b). *Software Quality Assurance Procedures for MELCOR*. Sandia National Laboratories

4.4.3 Software Quality-Related Issues or Concerns

A verifiable, written Software Design Document for MELCOR should be part of the written SQA Plan and Procedures for this software. Upgrades to the Model Description and other documentation can meet the intent of the Software Design Document for an interim period. However, in reconstituting the baseline for MELCOR, it is highly desirable that a new Software Design Document be developed. At a minimum, the Software Design Document should cover those modules that are used in LPF calculations.

4.4.4 Recommendations

Model descriptions in the MELCOR reference manual and other documentation and undocumented practices followed meet the intent of the software design document for the time being. Internal and independent testing of the existing code modules is believed to be robust. However, a software design report addressing the above table elements should be prepared. It is recommended that existing information on aerosol transport (theory, models, model results, tests, experiments, etc.) be gathered and consolidated and that the MELCOR LPF models be verified and validated against these within the context of the elements in Table 4.4-1.

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 the MELCOR transmittal confirm that the essential features of this criterion are met.
5.2	Implemented software was analyzed to identify and correct errors.	Yes.	Test problems exercising the model components are run prior to each release.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
5.3	The source code finalized during verification (this phase) was placed under configuration control.	Yes.	(SNL-1992b) is followed and configuration control is maintained on beta versions as well as release versions.
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.	Yes.	Copy of software and test case description are available. Not possible to trace to requirements and design documents which are lacking documentation.

4.5.2 Sources and Method of Review

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

4.5.3 Software Quality-Related Issues or Concerns

Not all criteria can be confirmed due to the lack of written records on implementation. However, based on available information, it is inferred that most of these requirements were met.

4.5.4 Recommendations

No recommendations related to this topical area are made.

4.6 Topical Area 6 Assessment: Testing Phase

This area corresponds to the requirement entitled Testing Phase in Table 3-3 of DOE (2003e). A Software Test Report has not been provided by the MELCOR software developers. Instead, a limited evaluation is performed applying Gauntt (2001), 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, especially Gauntt (2001), supports the satisfaction of this criterion.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
6.2	Testing demonstrated the capability of the software to produce valid results for test cases encompassing the range of permitted usage defined by the program documentation. Such activities ensured that the software adequately and correctly performed all intended functions.	Yes.	A series of test cases are run prior to release exercising most of the modules. Other testing is performed ad-hoc by the code authors.
6.3	Testing demonstrated that the computer program properly handles abnormal conditions and events as well as credible failures	Yes.	A series of test cases are run prior to release exercising most of the modules. Other testing is performed ad-hoc by the code authors.
6.4	Testing demonstrated that the computer program does not perform adverse unintended functions.	Yes.	A series of test cases are run prior to release exercising most of the modules. Other testing is performed ad-hoc by the code authors.
6.5	Test Phase activities were performed to assure adherence to requirements, and to assure that the software produces correct results for the test case specified. Acceptable methods for evaluating adequacy of software test case results included: (1) analysis with computer assistance; (2) other validated computer programs; (3) experiments and tests; (4) standard problems with known solutions; (5) confirmed published data and correlations.	Partial	A series of test cases are run prior to release exercising most of the modules. Other testing is performed ad-hoc by the code authors. Significant work has been performed to compare results to experiment. Current suite of test cases (Volume III) supplied with software includes commercial reactor and experimental facility examples. Documentation of requirements is lacking.
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.	Only partial record of testing is available. It is known that testing was conducted on MELCOR, and it is judged that the final version (1.8.5) performs as intended. However, resolution of unsuccessful cases is not possible to check, nor is traceability between test results and software requirements.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
6.7	Test procedures or plans specify the following, as applicable: (1) Required tests and test sequence, (2) Required range of input parameters, (3) Identification of the stages at which testing is required, (4) Requirements for testing logic branches, (5) Requirements for hardware integration, (6) Anticipated output values, (7) Acceptance criteria, (8) Reports, records, standard formatting, and conventions, (9) Identification of operating environment, support software, software tools or system software, hardware operating system(s) and/or limitations.	Partial.	A series of test cases are run prior to release exercising most of the modules. Other testing is performed ad-hoc by the code authors. No comprehensive detailed record of test procedures and plans was available. It can be inferred that this criterion was partially met. Complete verification was not possible due to lack of documentation.

4.6.2 Sources and Method of Review

SNL personnel were interviewed and documentation listed in Table 1-3 was reviewed.

4.6.3 Software Quality-Related Issues or Concerns

Lack of a test report for MELCOR forces the review to infer test case program results and outcome based on limited information. Volume 3 of the MELCOR 1.8.5 code manual (Gauntt, 2001) contains a portfolio of sample demonstration problems. These problems are a combination of experiment analyses, which illustrate code model performance against data, and full plant analyses showing MELCOR's performance on larger realistic problems. A few of these problems address, at least partially, aerosol transport, which is a key phenomenological area for LPF applications. While these studies promote confidence in the models for LPF applications, the documentation of these tests lack the necessary formality and comprehensiveness to address all components of the testing phase criterion.

4.6.4 Recommendations

A verifiable, written Test Report Document for MELCOR should be part of the written SQA Plan and Procedures for this software. Upgrades to the MELCOR software baseline will require that a Test Case Description and Report be completed. Test cases should include one or more example types that serve to demonstrate adequacy of the MELCOR software for LPF calculations that are representative of applications for DOE safety analysis. The Test Report and test phase documentation should address each of the above table elements.

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 MELCOR have been documented (Gauntt, 2000a; Gauntt, 2000b). Considered along with DOE-specific input preparation guidance in DOE (2003e), 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.	MELCOR models are described sufficiently (Gauntt, 2000a; Gauntt, 2000b).
7.2	User's manual or guide includes approved operating systems (for cases where source code is provided, applicable compilers should be noted).	Yes.	(Gauntt, 2000a; Gauntt, 2000b)
7.3	User's manual or guide includes description of the user's interaction with the software.	Yes.	(Gauntt, 2000a; Gauntt, 2000b)
7.4	User's manual or guide includes a description of any required training necessary to use the software.	Partial.	The MELCOR primer document discusses an approach a new user might take to become familiar with the code.
7.5	User's manual or guide includes input and output specifications.	Yes.	The User's manual (Gauntt, 2000a, Gauntt 2000b)
7.6	User's manual or guide includes a description of software and hardware limitations.	Yes.	The Reference Manual discusses the physics and models.
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.	Yes.	The code and manuals provide adequate diagnostics.
7.8	User's manual or guide includes information for obtaining user and maintenance support.	Yes.	The MELCOR website contains email and phone contact information.

4.7.2 Sources and Method of Review

Compliance with this requirement was evaluated by review of documentation listed in Table 1.3. SNL personnel were interviewed in January 2004.

4.7.3 Software Quality-Related Issues or Concerns

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

4.7.4 Recommendations

Recommendations related to this topical area are as follows:

- A simple training program would be useful. This could take several forms including a training manual, or interactive course. The novice user could be tasked with two to three simple problem types and walked through them with output information and explanation. The current sample case file could take on this function with expansion and concentration on LPF related elements.
- MELCOR limitations should be made more explicit in the User's Guide. Specific attention to limitations should be a focused topic and to the extent practical collected in one location.

4.8 Topical Area 8 Assessment: Acceptance Test

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

4.8.1 Criterion Specification and Result

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

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
8.1	To the extent applicable to the developer, acceptance testing includes a comprehensive test in the operating environment(s).	Yes.	Volume III (Gauntt 2001) and the electronic files provided allow the user to run a thorough test of the software. The sample problems should expand to provide one or more LPF specific cases.
8.2	To the extent applicable to the developer, acceptance testing was performed prior to approval of the computer program for use.	Yes.	Sample problem sets are run prior to release and checked. Errors or problems are corrected before release.

Criterion Number	Criterion Specification	Compliant	Summary Remarks
8.3	To the extent applicable to the developer, software validation was performed to ensure that the installed software product satisfies the specified software requirements. The engineering function (i.e., an engineering operation an item is required to perform to meet the component or system design basis) determines the acceptance testing to be performed prior to approval of the computer program for use.	Yes.	While documentation of requirements and comprehensive testing is lacking, the code is checked with a series of problems, and individual module testing is performed during development. Most new major modifications are compared against experiment and all are corrected before release.
8.4	Acceptance testing documentation includes results of the execution of test cases for system installation and integration, user instructions (Refer to Requirement 7 above), and documentation of the acceptance of the software for operational use.	Yes.	Volume III (Gauntt 2001) and the electronic files provided allow the user to run a thorough test of the software. Output for comparison is provided. Instructions are provided for installation.

4.8.2 Sources and Method of Review

Software package for code transmittal and documentation listed in Table 1.3 were reviewed. SNL personnel were interviewed in January 2004.

4.8.3 Software Quality-Related Issues or Concerns

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

4.8.4 Recommendations

No recommendations are made for this topical area.

4.9 Topical Area 9 Assessment: Configuration Control

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

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	Yes.	(SNL –1992b) provides details

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	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.		of required configuration control of the code and its related documentation.
9.2	Implementing procedures meet applicable criteria for configuration identification, change control and configuration status accounting.	Yes.	(SNL-1992b) provides details.

4.9.2 Sources and Method of Review

SNL personnel were interviewed in January 2004. (SNL-1992b) was reviewed and discussed.

4.9.3 Software Quality-Related Issues or Concerns

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

4.9.4 Recommendations

No recommendations are made for this topical area.

4.10 Topical Area 10 Assessment: Error Impact

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

4.10.1 Criterion Specification and Result

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

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
10.1	The 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.	Yes.	The process used for monitoring errors and user feedback on MELCOR is defined in (SNL-1992b). This was formerly strictly followed. It continues to be followed, but less rigidly than before, in part, because of funding considerations.
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.	Partial.	Some guidance is given in (SNL-1992b). Judgment is used by the authors to determine the severity of the error. Formal specifications to help with this judgment are lacking.
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.	Guidance is given in (SNL-1992b) Errors and defects are handled by logging them and including updates in the next release. Notification is lacking formality usually associated with a safety related code. Procedures state notification depends on funding. NRC as the current sponsor and SNL define MELCOR as a research code. The reporting scheme currently conforms to this definition.
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.	Guidance is given in (SNL-1992b).
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.	Some guidance is given in (SNL-1992b). In practice, this may be accomplished but is not automatic and is left to the judgment of the authors.
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	No.	No information was available to support that this occurs formally. Rather consistency of personnel and experience

Criterion Number	Criterion Specification	Compliant	Summary Remarks
	impacts previous development activities		are used to the extent this is accomplished.
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.	No.	Errors and defects are handled by logging them and including updates in the next release. Notification is lacking formality. Procedures state notification depends on funding. NRC as the current sponsor and SNL define MELCOR as a research code. The reporting scheme conforms to this definition.

4.10.2 Sources and Method of Review

SNL personnel were interviewed in January 2004. SNL has an informal Software Reporting system. The MELCOR website has a link to send an e-mail to MELCOR technical staff. Staff indicated that email is the primary means by which defects are reported. Through the FAQ link on the MELCOR website, users can read about problems other users have reported and see the response of the MELCOR technical staff. The effectiveness or timeliness of this system, however, is difficult to judge. Under the FAQ link, the MELCOR technical staff relays user-reported problems, discuss the causes of error messages, and provide tips to avoid discovered problems until a patch or new version is distributed. As of January 2004, six problems were addressed at the FAQ link. None have been identified as having any significant impact on LPF results.

4.10.3 Software Quality-Related Issues or Concerns

While an informal Software Reporting system process is institutionalized at SNL, its effectiveness can not be established. The authors make concerted effort to record emails they receive, and log the information as it comes in internally. Notification to users of defects on a timely basis, close out with the defect reporter, and formal impact determination are in need of improvement.

4.10.4 Recommendations

As part of the new software baseline for MELCOR, a comprehensive Software Error Notification and Corrective Action process should be provided. Expanded use of the MELCOR website or its equivalent is suggested to provide timely reporting of user issues, errors and defects. It may also provide software news, suggested strategies for resolving software problems, and general communications. Timely, formal user notification of errors or defects should be addressed.

4.11 Training Program Assessment

Current MELCOR training opportunities are limited and not well publicized. Comprehensive training on a more frequent basis would be beneficial.

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

In May 2004 the MELCOR Code Application Program (MCAP) group is planning to meet near Washington DC. The first day of this meeting is closed to non-members. Potential exists to add training for MELCOR, both general, or specific to LPF, at the end of this meeting.

Training could result in MELCOR LPF certification/qualification. This level of user proficiency could be measured by demonstrating competency through a written exam and software execution of a set of test cases. Ideally, this could be accomplished through formal course attendance or through a self directed (self-study) process.

4.12 Software Improvements and New Baseline

The minimum remedial program required to yield the new software baseline for MELCOR was discussed earlier as part of Table 1.1. Included are upgrades to software documents that constitute the baseline for software, including:

- Updated Software Quality Assurance Plan
- Software Requirements Document (Specific to LPF)
- Software Design Document (Specific to LPF)
- Test Case Description and Report (Specific to LPF)
- Updated Software Configuration and Control
- Updated Error Notification and Corrective Action Report Procedure, and
- Updated User's Manual.

The SNL procedural guide NP-19 implements an earlier version of Subpart 2.7 to NQA-1, specifically NQA-2a-1990. Application of this procedure was assessed for the SNL MACCS2 code with the result being the minimum set of actions as documented in Bixler (2000) and shown below in Table 4.12-1. Column "SNL NP 19-1 (Bixler)". Application of this procedure to MELCOR can be expected to result in a similar set of actions as specified in the column labeled "Corresponding Recommended Steps from this GAP analysis".

While not exactly matching up with the recommendations proposed in this gap analysis, the SNL proposed program is similar to the requirements outlined in this report. Furthermore, the estimates are based on SNL 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 was estimated to require 1.5 full-time equivalent years to complete. The requirements are matched against the requirements earlier, in Table 4.12-1. The overall level of effort, 1.5 FTE-years is rounded up to approximately 2 FTE-years as the final estimate for resource allocation to perform the upgrades required to compensate for MELCOR's

known SQA gaps. This is a very rough estimate based on this comparison, extrapolating from MACCS to MELCOR and considering the differences. It assumes there would not be major defects found as the program is completed and that existing information would be adequate to complete verification and validation of the LPF models. Long term, maintenance funding will be required for activities such as defect reporting, coordinated update testing as NRC makes changes in the future, and minor SQA administrative duties.

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

Topic No.	Topic: ASME NQA-1-2000 Require-ments	Level B Existing Software (Topic Applied?)	GAP Report Section No.	SNL NP 19-1 Steps (Bixler)	Compliance Steps in this GAP Document, DOE (2003e)
1	Software Classifi-cation	Yes	4.1	None	None
2	SQA Proce-dures/ Plans	Yes	4.2	Create a Primitive Baseline (PB) document to establish the SQA status of the existing code	Update SQA plan
3	Dedica-tion	No[4]	–	–	–
4	Evalua-tion	No[4]	–	–	–
5	Require-ments	Yes	4.3	Write a Software Requirements Document (SRD)	Write a Software Requirements Document (SRD)
6	Design Phase	Yes	4.4	None	Write a Design Document
7	Imple-menta-tion Phase	Yes	4.5	Create an Implementation Document (ID) to describe the process of generating the executable software modules	Create an Implementation Document (ID) to describe the process of generating the executable software modules

[4] Topic evaluated as not significantly relevant to safety analysis toolbox codes.

8	Testing Phase	Yes	4.6	Establish a Verification and Validation Plan (VVP) based on the SRD; Generate a Validation Document (VD), to measure the performance of the software against the criteria specified in the VVP	Establish a Verification and Validation Plan (VVP) based on the SRD; Generate a Validation Document (VD), to measure the performance of the software against the criteria specified in the VVP
9	User Instruc-tions	Yes	4.7	Update, the User's Manual (UM)	Update, the User's Manual (UM)
10	Accept-ance Test	Yes	4.8	Perform Installation and Checkout (I&C) to verify correct installation on all supported platforms	None (normally done for MELCOR))
11	Opera-tion and Mainten-ance	No[4]	–	–	–
12	Config-uration Control	Yes	4.9	Implement a Software Configuration Control System (CC)	Update Software Configuration Control System (CC)
13	Error Impact	Yes	4.10	Implement a Software Problem Reporting System (SPR)	Update Software Problem Reporting System (SPR)
14	Access Control	No[4]	–	–	–

5.0 Conclusions

The gap analysis for Version 1.8.5 of the MELCOR 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), five requirements are met at an acceptable level, i.e., *Software Classification (1), Implementation Phase (5), User Instruction s(7)*, *Acceptance Test (8),* and *Configuration Control (9)*. Five topical quality areas were not met satisfactorily. Improvement actions are recommended for MELCOR to fully meet SQA criteria for the remaining five requirements.

It was determined that the MELCOR code as applied to LPF calculations 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 MELCOR for supporting safety analysis. Informed use of the software can be assisted by the current set of MELCOR reports (refer to Table 1-3), and the code guidance report for DOE safety analysts, *MELCOR Computer Code Application Guidance for Leak Path Factor in Documented Safety Analysis*, (DOE, 2004). Furthermore, while SQA improvement actions are recommended for MELCOR as applied to LPF calculations, no evidence has been found of programming, logic, or other types of software errors in MELCOR 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 MELCOR software improvement actions be taken, especially:

1. Correcting known defects in the SQA process
2. Upgrading existing SQA documentation
3. Providing training on a regular basis, and
4. Revising and developing new software documentation.

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

- Updated Software Quality Assurance Plan
- Software Requirements Document (Specific to LPF)
- Software Design Document (Specific to LPF)
- Test Case Description and Report (Specific to LPF)
- Updated Software Configuration and Control
- Updated Error Notification and Corrective Action Report Procedure, and
- Updated User's Manual.

Approximately two full-time equivalent years is conservatively estimated to upgrade MELCOR software to be compliant with NQA-1-based requirements for existing software. 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 MELCOR is planned for release in the future. It is recommended that this version be evaluated upon issue relative to the software improvement and baseline recommendations, and the full set of SQA criteria discussed in this report. If this version is found to be satisfactory, it should replace Version 1.8.5 as designated version of the software for the toolbox.

Approximately one FTE-month per year would be needed to maintain a web-based error notification and corrective action process for MELCOR (Section 4.10). However, such a process has not been defined in depth for MELCOR 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)
CFD	Computational Fluid Dynamics
CFR	Code of Federal Regulations
CSARP	Cooperative Severe Accident Research Program
DCF	Dose Conversion Factor
DIR	Defect Investigation Report
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.

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

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

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

Nuclear Facility — A reactor or a nonreactor nuclear facility where an activity is conducted for or on behalf of DOE and includes any related area, structure, facility, or activity to the extent necessary to ensure proper implementation of the requirements established by 10 CFR 830. [10 CFR 830]

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

Safety Analysis and Design Software – Computer software that is not part of a structure, system, or component (SSC) but is used in the safety classification, design, and analysis of nuclear facilities to ensure the proper accident analysis of nuclear facilities; the proper analysis and design of safety SSCs; and, the proper identification, maintenance, and operation of safety SSCs. [DOE O 414.1B]

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

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

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

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

7.0 References

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.

Bixler, N. (2000). *Proposal to Resolve QA Deficiencies in MACCS2*, Memorandum to D. Chung (DOE/DP), Sandia National Laboratories, Albuquerque, NM (2000).

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). MELCOR Computer Code Application Guidance for Leak Path Factor in Documented Safety Analysis, (May 2004).

East, J. M. (1998) 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).

Gauntt, R. O. (2000a) et al. *MELCOR Computer Code Manuals, Vol. 1: Primer and Users' Guide*, Version 1.8.5, NUREG/CR-6119 Rev. 2, SAND2000-2417/1, May 2000.

Gauntt, R. O. (2000b) et al. *MELCOR Computer Code Manuals, Vol. 2: Reference Manuals*, Version 1.8.5, NUREG/CR-6119 Rev. 2, SAND2000-2417/2, May 2000.

Gauntt, R. O. (2001) et al. *MELCOR Computer Code Manuals, Vol. 3: Demonstration Problems*, Version 1.8.5, NUREG/CR-6119 Rev. 0, SAND2001-0929P, May 2001.

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

SNL (1992b). *Software Quality Assurance Procedures for MELCOR*. Sandia National Laboratories, Albuquerque, NM, Revision 1.2, August 2, 1992.

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

SNL (2001). *5th MELCOR User's Workshop*, Sandia National Laboratories, Bethesda, MD, May 10th – 15th, 2001.

SNL (2003). *Software Requirements*, Revision 10, Nuclear Waste Management Procedure, NP 19-1, Waste Isolation Pilot Plant, (May 2003).

Appendices

Appendix	Subject
A	Software Information Template

APPENDIX A. SOFTWARE INFORMATION TEMPLATE

The following is a condensed version of the information request sent to the MELCOR code developer in October 2003. *(Note: This information is provided to give the reader of this report, an idea of the information requested to complete the gap analysis for MELCOR. Detailed information in response was not filled in. See Section 1.6. Instead, the contacts and the SQA evaluators used the form as a guide for continual discussion throughout the gap analysis for MELCOR.*

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 MELCOR code.

Table 2. Summary Description of Subject Software

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

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

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

Table 3. Point of Contact for Form Completion

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

1. **Software Quality Assurance Plan**

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

1.a For this software, identify the governing Software Quality Assurance Plan (SQAP)?
[Please submit a PDF of the SQAP, or send hard copy of the SQAP[5]]

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:

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

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

2. Software Requirements Description

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

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

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

Guidance for Software Requirements Documentation:

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

3. Software Design Documentation

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

The SDD should provide the following:

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

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

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

Guidance for Software Design Documentation:

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

4. Software User Documentation

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

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

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

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

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

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

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. <u>Software Verification &Validation Documentation (Includes Test Reports)</u>

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 correctic
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. <u>Resource Estimates</u>

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.