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

Software Quality Assurance Improvement Plan: GENII 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 document provides an evaluation of the Software Quality Assurance (SQA) attributes of GENII, a radiological dispersion computer code, relative to established requirements. This evaluation, a "gap analysis", is performed to meet commitment 4.2.1.3 of the Department of Energy's Implementation Plan to resolve SQA issues identified in the Defense Nuclear Facilities Safety Board Recommendation 2002-1. Both versions of the GENII code (1.485 and 2.0) are addressed.

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

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

INTENTIONALLY BLANK

REVISION STATUS

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

INTENTIONALLY BLANK

CONTENTS

INTENTIONALLY BLANK

TABLES

FIGURES

Software Quality Assurance Improvement Plan: GENII Gap Analysis

EXECUTIVE SUMMARY

The Defense Nuclear Facilities Safety Board (DNFSB) issued Recommendation 2002-1 on *Quality Assurance for Safety-Related Software* in September 2002 (DNFSB 2002). The Recommendation identified a number of quality assurance issues for software used in the Department of Energy (DOE) facilities for analyzing hazards and designing and operating controls that prevent or mitigate potential accidents. The development and maintenance of a collection, or "toolbox," of high-use, Software Quality Assurance (SQA)-compliant safety analysis codes is one of the major improvement actions discussed in the *Implementation Plan for Recommendation 2002-1 on Quality Assurance for Safety Software at Department of Energy Nuclear Facilities* (DOE 2003a). 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 GENII software, for radiological dispersion and consequence analysis, is one of the codes designated for the toolbox. To determine the actions needed to bring the GENII code into compliance with the SQA qualification criteria, and develop an estimate of the resources required to perform the upgrade, the Implementation Plan has committed to sponsoring a code-specific gap analysis document. The gap analysis evaluates the software quality assurance attributes of GENII against identified criteria.

The balance of this document provides the outcome of the GENII 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). For GENII 1.485, of the ten general topical quality areas that were evaluated for software developers, nine met the criteria fully, and one failed to meet the criteria. For GENII 2.0, of the ten general topical quality areas, two met the criteria fully, five met the criteria partially, and three failed to meet the criteria. Recommendations are given for each of the topical areas in Section 4.0. The GENII code was evaluated to determine if the code, as it currently stands, meets the intended function for the code in the context as described in the scope of this gap analysis. When the code is run for the intended applications, as detailed in the code guidance document, *GENII Computer Code Application Guidance for Documented Safety Analysis*, (DOE 2004), it is judged that GENII 1.485 will meet its intended function, but GENII 2.0 will not. Therefore, only GENII 1.485 can be recommended for DSA use at this time. Note, however, that the GENII guidance document (DOE 2004) specifies that GENII 1.485 should be run either on a DOS-based computer, or in a DOS window of a Windows-95 or –98 based computer, not in a DOS window of a Windows XP based computer where problems may be encountered.

It is estimated that nearly ten full-time equivalent (FTE) months would be required to perform all SQA upgrade tasks identified in Section 4.0 of this report for GENII 2.0.

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), nine requirements are met at an acceptable level for GENII 1.485, (items 1-9). Improvement actions are recommended for GENII 1.485 to fully meet the requirement for *Error Impact* (item 10). For GENII 2.0, of the ten primary SQA requirements for existing software at the Level B classification (important for safety analysis but whose output is not applied without further review), two requirements are met at an acceptable level, i.e., *Software Classification* (1) and *Configuration Control* (9). Improvement actions are recommended for

GENII 2.0 to fully meet the requirement for five that are partially met, i.e., *SQA Procedures and Plans* (2), *Requirements Phase* (3), *Design Phase* (4), *Implementation Phase* (5), and *User Instructions* (7) and for the remaining three, *Testing Phase* (6), *Acceptance Test* (8), and *Error Impact* (10). This evaluation outcome is deemed acceptable because: (1) GENII 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 GENII is limited to those analytic applications for which the software is intended.

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

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

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

A new software baseline set of documents is recommended for GENII 2.0 to demonstrate completion of item 4 (above), revise software documentation. The list of baseline documents for revision includes:

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

While completion of the GENII 2.0 development is encouraged, current DOE DSA support should be through the earlier code version, GENII 1.485, run on a DOS-based computer or a Windows-95 or -98 based computer. No evidence was found of software-induced errors in GENII 1.485 that have led to non-conservatisms in nuclear facility operations or in the identification of facility controls.

1.0 Introduction

This document reports on the results of a gap analysis for the GENII computer code. Both versions of the code (1.485 and 2.0) are considered. The intent of the gap analysis is to determine the actions needed to bring the specific software into compliance with established Software Quality Assurance (SQA) criteria. A secondary aspect of this report is to develop an estimate of the level of effort required to upgrade GENII 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), were designated by DOE for the toolbox (DOE/EH, 2003). It is found that these codes provide 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, GENII will likely require some degree of quality assurance improvement before meeting current SQA standards. The analysis documented herein is an evaluation of GENII, both versions 1.485 and 2.0, relative to current software quality assurance criteria. It assesses the extent of the deficiencies, or gaps, to provide DOE and the software developer the extent to which minimum upgrades are needed. The overall assessment is therefore termed a "gap" analysis.

1.2 Evaluation of Toolbox Codes

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

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

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

Early in the SQA evaluation program, it was anticipated that each toolbox code owner would provide input information on the SQA programs, processes, and procedures used to develop their software. However, most of the designated toolbox software, including GENII, 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 GENII development program. While independent of the code developer, the SQA evaluators responsible for GENII 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 will provide information to DOE, code developers, and code users.

DOE will see the following benefits:

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

Each code developer will be provided:

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

DOE safety analysts and code users will benefit from:

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

1.4 Scope

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

1.6 Methodology for Gap Analysis

The gap analysis for GENII is based on the criteria as described in *Software Quality Assurance Plan and Criteria for the Safety Analysis Toolbox Codes* (DOE 2003e). The overall methodology for the gap analysis is summarized in Table 1-1. The gap analysis utilizes ten of the fourteen topical areas listed in DOE (2003e) related to software quality assurance to assess the GENII software. The ten areas are those particularly applicable to the software development, specifically: (1) Software Classification, (2) SQA Procedures/Plans, (5) Requirements Phase, (6) Design Phase, (7) Implementation Phase, (8) Testing Phase, (9) User Instructions, (10) Acceptance Test, (12) Configuration Control, and (13) Error Impact. Each area, or requirement, is assessed individually in Section 4.

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

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.,

[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
	• Software Requirements Document • Software Design Document • Test Case Description and Report • Software Configuration and Control Document • Error Notification and Corrective Action Report, and • User's Instructions (alternatively, a User's Manual), Model Description (if this information has not already been covered). c. Identify documents that are acceptable from SQA perspective. Note inadequate documents as appropriate.
3. Software Product Technical/ Functional Requirements	a. Review requirements documentation to determine if requirements support intended use in Safety Analysis. Document this determination in gap analysis document. b. Review previously conducted software testing to verify that it sufficiently demonstrated software performance required by the Software Requirements Document. Document this determination in the gap analysis document.
4. Testing	a. Determine whether past software testing for the software being evaluated provides adequate assurance that software product/technical requirements have been met. Obtain documentation of this determination. Document this determination in the gap analysis report. b. (Optional) Recommend test plans/cases/acceptance criteria as needed per the SQAP if testing not performed or incomplete.
5. New Software Baseline	a. Recommend remedial actions for upgrading software documents that constitute baseline for software. Recommendations can include complete revision or providing new documentation. A complete list of baseline documents includes: • SQA Plan • Software Requirements Document • Software Design Document • Test Case Description and Report • Software Configuration and Control • Error Notification and Corrective Action Report, and • User's Instructions (alternatively, a User's Manual) b. Provide recommendation for central registry as to minimum set of SQA documents to constitute new baseline per the SQAP.
6. Training	a. Identify current training programs provided by developer. b. Determine applicability of training for DOE facility safety analysis.
7. Software Engineering Planning	a. Identify planned improvements of software to comply with SQA requirements. b. Determine software modifications planned by developer. c. Provide recommendations from user community. d. Estimate resources required to upgrade software.

1.7 Summary Description of Software Being Reviewed

The gap analysis was performed on both versions of the GENII code (i.e., Version 1.485 [Napier, 1988a, 1988b, 1988c] and Version 2.0 [Napier, 1995, 2002a, 2002b, 2003]). Although the earlier version (1.485) is the one recommended for use in current DSAs, the later version (2.0) is also evaluated, because the improvements recommended here, if implemented, would allow it to be used in DSAs in the future. In the following discussion, RSICC refers to the Radiation Safety Information Computational Center at Oak Ridge, TN.

The set of documents reviewed as part of the gap analysis are listed in Table 1-2.

Table 1-2 — Software Documentation Reviewed for GENII

No.	Information	
1.	Reference:	B. A. Napier, R. A. Peloquin, D. L. Strenge, and J. V. Ramsdell, *GENII – The Hanford Environmental Radiation Dosimetry Software System. Volume 1: Conceptual Representation.* PNL-6584, December 1988. (Napier, 1988a)
	Remarks:	Documentation provided by RSICC in .pdf format
2.	Reference:	B. A. Napier, R. A. Peloquin, D. L. Strenge, and J. V. Ramsdell, *GENII – The Hanford Environmental Radiation Dosimetry Software System. Volume 2: User's Manual*, PNL-6584, November 1988. (Napier, 1988b)
	Remarks:	Documentation provided by RSICC in .pdf format
3.	Reference:	B. A. Napier, R. A. Peloquin, D. L. Strenge, and J. V. Ramsdell, *GENII – The Hanford Environmental Radiation Dosimetry Software System. Volume 3: Code Maintenance Manual*, PNL-6584, September 1988. (Napier, 1988c) Only the table of contents is available (included as part of the .pdf file of Volumes 1 and 2). Bruce Napier has one of the few copies of the entire document (Volume 3), which is about 1,500 pages long, but a copy was not available for this gap analysis.
	Remarks:	Table of contents in .pdf format provided by RSICC.
4.	Reference:	B. A. Napier, J. V. Ramsdell, and D. L. Strenge, *Software Requirements Specifications for Hanford Environmental Dosimetry Coordination Project*, Draft Report, prepared for review by the EPA Office of Radiation and Indoor Air, May 1995. (Napier, 1995)
	Remarks:	Documentation provided by Bruce Napier.
5.	Reference:	B. A. Napier, *GENII Version 2 User's Guide* (Napier, 2002a)
	Remarks:	Downloaded from PNNL website
6.	Reference:	B. A. Napier, D. L. Strenge, J. V. Ramsdell, Jr., P. W. Eslinger, and C. Fosmire, *GENII Version 2 Software Design Document* (Napier, 2002b)
	Remarks:	Downloaded from PNNL website
7.	Reference:	B. A. Napier, *GENII Version 2 Example Calculation Descriptions* (Napier, 1999a)
	Remarks:	Documentation on CD from EFCOG training class, June 1999
8.	Reference:	B. A. Napier and L. Staven, *GENII Version 2 Training Power Point Slides* (Napier, 1999b)
	Remarks:	Documentation on CD from EFCOG training class, June 1999

Table 1-2 — Software Documentation Reviewed for GENII (continued)

No.	Information	
9.	Reference:	B. A. Napier, *Getting Started with GENII Version 2* (Napier, 2003)
	Remarks:	Downloaded from EPA/NESHAPs website
10.	Reference:	B. A. Napier, E-mail communications with K. R. O'Kula and Vern Peterson
	Remarks:	Provided in Appendix A
11.	Reference:	W. E. Joyce, Telephone conversation with V. L. Peterson
	Remarks:	Provided in Appendix A
12.	Reference:	Publications supporting GENII Benchmarking and V&V
	Remarks:	Provided in Appendix B

2.0 Assessment Summary Results

2.1 Criteria Met

For GENII 1.485, of the applicable ten general topical quality areas, nine met the criteria fully, and one failed to meet the criteria. An exception was found in the area of Error Impact. An area for GENII 1.485 improvement is to create and follow a formal error reporting and corrective action process. For GENII 2.0, of the ten general topical quality areas, two met the criteria fully, five met the criteria partially, and three failed to meet the criteria. Exceptions were found in the areas of Testing Phase, Acceptance Test, Error Impact, and partially in the areas of SQA Procedures and Plans, Requirements Phase, Design Phase, Implementation Phase, and User Instructions.

The areas that should be addressed for improvement actions are listed in Section 2.2 (Exceptions to Requirements). Detail on the evaluation process relative to the requirements, and the criteria applied, are found in Section 4.

2.2 Exceptions to Requirements

Some of the more important exceptions to criteria found for GENII 2.0 are listed below in Table 2-1. No similar list is needed for GENII 1.485. 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 for GENII 2.0

No.	Criterion	Reason Not Met	Remedial Action(s)
1.	Testing Phase	Testing not yet complete	Document all testing of GENII 2.0
2.	Acceptance Test	Testing not yet complete	Develop and document acceptance criteria for GENII 2.0 and document acceptance testing.
3.	Error Impact	A formal error reporting and corrective action procedure is not followed.	Create and follow a formal error reporting and corrective action process (applies to GENII 1.485 as well)

2.3 Areas Needing Improvement

The gap analysis identified a number of improvements that could be made related to the code and its quality assurance. Some of the important ones are listed in Table 2-2.

Table 2-2 — Summary of Important Recommendations for GENII

No.	Recommendation
1.	Establish and follow formal review schedules for GENII 2.0.
2.	Make GENII 2.0 code listings available upon completion and final testing of code.
3.	Correct the user documentation (see Section 4.7.4) and the bugs in the user interface for GENII 2.0 (see Criterion 9.6).
4.	Run GENII 1.485 only on a DOS-based computer, or in a DOS window of a Windows-95 or –98 based computer. Problems may be encountered when GENII 1.485 is run under Windows-XP. It is recommended that GENII 1.485 be recompiled with a FORTRAN compiler that is compatible with Windows-XP, tested, and released as an updated version.
5.	Modify GENII 2.0 to make it easy for the user to determine 95th percentile consequences at the site boundary and at a user-selected collocated worker distance (for example, 100 m).
6.	Assemble the existing "software change packets" for GENII 1.485 into a document to verify that changes to the code followed a logical and verifiable process.

2.4 Areas Not Assessed and Any Limitations of Gap Analysis

All areas were assessed for this gap analysis. Some areas were found to be more difficult to assess than others, depending upon the level of detail provided in the documentation. However, no limitations were imposed on the gap analysis.

2.5 Conclusion Regarding Software's Ability to Meet Intended Function

The GENII 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, *GENII Computer Code Application Guidance for Documented Safety Analysis* (DOE 2004), and also utilizing information from documentation available (Table 1-2), it is judged that GENII 1.485 will meet the intended function, but GENII 2.0 will not. Therefore, only GENII 1.485 can be recommended for DSA use at this time. Note, however, that the GENII guidance document (DOE 2004) specifies that GENII 1.485 should be run on a DOS-based computer, or in a DOS window of a Windows-95 or –98 based computer, not in a DOS window of a Windows-XP based computer where problems may be encountered.

The primary remedial actions required for GENII 2.0 include the following:

(1) Modify the software so that the user can determine the 95th percentile doses at the site boundary in all sectors
(2) Improve the user documentation
(3) Create an error-reporting and corrective action procedure, including its documentation
(4) Complete code testing and document it
(5) Create and implement a code maintenance procedure.

3.0 Lessons Learned

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

Table 3-1 — Lessons Learned

No.	Lesson
1.	Changing criteria in SQA standards over the years can render codes non-compliant that were once compliant.
2.	Although the author of a code may intend the code to be compliant with SQA standards, the standards may present sufficient complexity so that some requirements are not met in total.
3.	Development of software that is compliant with SQA standards can be a costly and laborious endeavor, especially if it is back-fit to the software, instead of being a parallel requirement during software development. If funding for the project is meager, SQA will probably not be followed as closely as may have been intended originally. Completion of the code development may take precedence over SQA measures.
4.	Changing sponsors may impact the SQA pedigree of software. This situation can arise especially if more recent software development was driven by other, non-SQA requirements than were present originally. The current version of the code has been developed for Environmental Protection Agency (EPA)/National Emission Standards for Hazardous Air Pollutants (NESHAPS), while original versions of the code were funded out of the PNNL budget.

4.0 Assessment Detailed Results

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

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*

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

Subsection (This Report)	Corresponding Entry Table 3-3 from DOE (2003e)	Requirement	ASME NQA-1 2000 Section/Consensus Standards
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

This topical area is "required" for both GENII 1.485 and 2.0. Table 4.1-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

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 for both	The documentation from the developer makes it clear that both GENII 1.485 and 2.0 are Level B software.

4.1.2 Sources and Method of Review

All of the documentation listed in Table 1-2 has been reviewed with attention to "Software Classification," except for Item 12 (see Appendix B).

4.1.3 Software Quality-Related Issues or Concerns

There are no other SQA-related issues or concerns in "Software Classification."

4.1.4 Other Areas for Improvement

No areas of improvement in "Software Classification" have been noted.

4.1.5 Recommendations

This requirement is met. There are no recommendations related to this Topical Area.

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 the DOE SQA plan (DOE 2003e). It deals with the planning efforts prior to code development.

4.2.1 Criterion Specification and Result

This topical area is "required" for both GENII 1.485 and 2.0. Table 4.2-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
2.1	Procedures/plans for SQA (SQA Plan) have identified organizations responsible for performing work, independent reviews, etc.	Yes for both	Pacific Northwest National Laboratory (PNNL) (formerly Pacific Northwest Laboratory [PNL]) is responsible for performing the work and providing for independent reviews (Napier, 1988a) and Napier (1995)

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
2.2	Procedures/plans for SQA (SQA Plan) have identified software engineering methods.	Yes for both	The software engineering methods are discussed in Napier (1988a) and Napier (1995)
2.3	Procedures/plans for SQA (SQA Plan) have identified documentation to be required as part of program.	Yes for both	Required documentation is discussed in Napier (1988a) and Napier (1995)
2.4	Procedures/plans for SQA (SQA Plan) have identified standards, conventions, techniques, and/or methodologies that shall be used to guide the software development, methods to ensure compliance with the same.	Yes for both	The standards, conventions, techniques, and/or methodologies that were used to guide code development are discussed in Napier (1988a) and Napier (1995).
2.5	Procedures/plans for SQA (SQA Plan) have identified software reviews and schedule.	Yes for 1.485. No for 2.0.	Napier (1988a) discusses two formal review periods for GENII 1.485. No similar discussion is in the GENII 2.0 documentation.
2.6	Procedures/plans for SQA (SQA Plan) have identified methods for error reporting and corrective actions.	Yes for 1.485. No for 2.0	Napier (1988b) discusses how to report errors and request upgrades. An informal method is used for GENII 2.0.

Additional Detail

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

Criterion 2.1 — The GENII 1.485 system was developed under the direction of the DOE office at Hanford for use by nuclear safety analysts. Potential user groups were identified and representatives of these groups were then selected to form a committee to specify the software requirements. Other groups were identified to provide reviews of the design and perform independent testing. The documentation describes these groups by their functions and the names of individual members are given in the "Acknowledgements" section. The organization selected to perform the work was the PNL (now PNNL). The GENII 2.0 system was developed with funding from the EPA. It incorporates much of the code developed for GENII 1.485 but was developed for use by the EPA in Environmental Impact Statements (EISs). The various groups for review and testing are mentioned in Napier (1995), which is the SQA plan for GENII 2.0.

Criterion 2.2 — An appendix to the GENII 1.485 volume 1 (Napier, 1988a) is a detailed system-requirements document. In it, software engineering methods are discussed. For GENII 2.0, the system requirements are given in Napier (1995), which discusses software engineering. (However, the word "engineering" is not used in either document.)

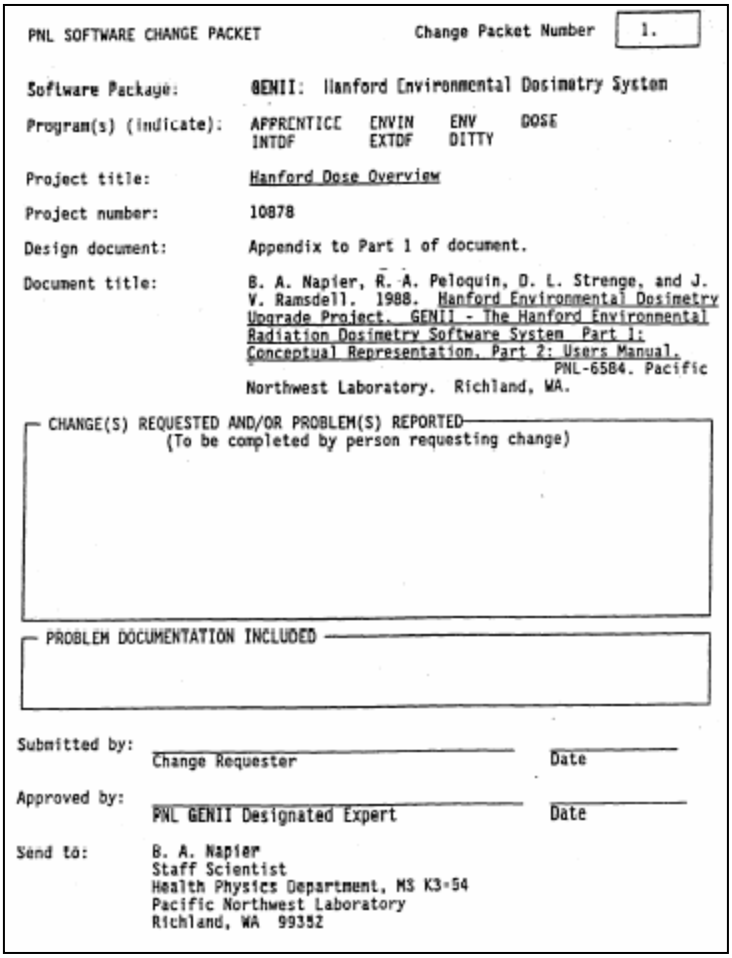
Criterion 2.3 — The GENII 1.485 documentation (Napier, 1988a, 1988b) identified several required documents, including requirements for the overall system, design, implementation, testing, user manual, and maintenance. Likewise, Napier (1995) discusses the planned documentation for GENII 2.0.

Criterion 2.4 — Napier (1988a) and Napier (1995) discuss the standards, conventions, techniques, and/or methodologies to be used to guide code development. Napier (1988a) was

prepared during and after the development of GENII 1.485 and is, thus, more detailed than Napier (1995), which was prepared before the development of GENII 2.0.

Criterion 2.5 — External peer reviews of GENII 1.485 were conducted during the weeks beginning September 14, 1987 and February 1, 1988. This was followed by a formal acceptance of the code upon completion of the documentation packages for the user. Review schedules are not discussed in the GENII 2.0 documentation.

Criterion 2.6 — A formal error-reporting methodology was used for GENII 1.485. A copy of the reporting form is shown in Figure 4-1. For GENII 2.0, error reporting is informal, as evidenced by e-mail from Napier (see Appendix A) that includes the statement "I only have a few beta users; they let me know when it's broke and I fix it for them."



PNL SOFTWARE CHANGE PACKET — Change Packet Number: 1.

Software Package: GENII: Hanford Environmental Dosimetry System

Program(s) (Indicate): APPRENTICE ENVIN ENV DOSE INTDF EXTDF DITTY

Project title: Hanford Dose Overview

Project number: 10878

Design document: Appendix to Part 1 of document.

Document title: B. A. Napier, R. A. Peloquin, D. L. Strenge, and J. V. Ramsdell. 1988. Hanford Environmental Dosimetry Upgrade Project. GENII - The Hanford Environmental Radiation Dosimetry Software System Part 1: Conceptual Representation, Part 2: Users Manual. PNL-6584. Pacific Northwest Laboratory. Richland, WA.

CHANGE(S) REQUESTED AND/OR PROBLEM(S) REPORTED
(To be completed by person requesting change)

PROBLEM DOCUMENTATION INCLUDED

Submitted by: _____ Change Requester — Date _____

Approved by: _____ PNL GENII Designated Expert — Date _____

Send to: B. A. Napier
Staff Scientist
Health Physics Department, MS K3-54
Pacific Northwest Laboratory
Richland, WA 99352

Figure 4-1 — Error reporting / update request form for GENII 1.485

4.2.2 Sources and Method of Review

All of the documentation listed in Table 1-2 has been reviewed with attention to "SQA Procedures and Plans," except for Item 12 (see Appendix B).

4.2.3 Software Quality-Related Issues or Concerns

Review schedules and a formal error reporting and corrective action methodology needs to be implemented for GENII 2.0.

4.2.4 Other Areas for Improvement

No other areas of improvement are noted.

4.2.5 Recommendations

Recommendations related to this topical area are provide in Table 4.2-2.

Table 4.2-2 — Recommendations for SQA Procedures and Plans Topic

Recom-mendation Number	Relates to Table 4.2-1 Criterion Number(s)	Recommendation	Est. FTE to Complete	Est. Calendar Duration
2.1	2.6	Implement a Formal Error Report (FER) and handling methodology for GENII 2.0. This is not required for GENII 1.485.	One FTE week	Two weeks
2.2	2.5	Establish formal review schedules for GENII 2.0.	One FTE day	One week

4.3 Topical Area 3 Assessment: Requirements Phase

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

4.3.1 Criterion Specification and Result

This topical area is "required" for both GENII 1.485 and 2.0. Table 4.3-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
3.1	Software requirements for the subject software have been established.	Yes for both	Software Requirements are in: 1.485: Napier (1988a) appendix 2.0: Napier (1995)
3.2	Software requirements are specified, documented, reviewed, and approved.	Yes for both	1.485: Software specifications, review, and approval are in Napier (1988a) and its appendix. 2.0: Requirements in Napier (1995). Review and approval implied by Napier (2002b).
3.3	Requirements define the functions to be performed by the software and provide detail and information necessary to design the software.	Yes for both	Detailed functional requirements are defined in: 1.485: Napier (1988a) appendix 2.0: Napier (1995)
3.4	A Software Requirements Document, or equivalent, defines requirements for functionality, performance, design inputs, design constraints, installation considerations, operating systems (if applicable), and external interfaces necessary to design the software.	Yes for both	Detailed functional requirements are defined in the System Requirements documents: 1.485: Napier (1988a) appendix 2.0: Napier (1995)
3.5	Acceptance criteria are established in the software requirements documentation for each of the identified requirements.	Yes for 1.485. Partial for 2.0	1.485: Napier (1988b, 1988c) 2.0: Acceptance criteria are not specifically described but are implied by testing requirements

Additional Detail

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

Criteria 3.1 and 3.2 — GENII 1.485 was developed by means of tasks designed to provide a state-of-the-art, technically peer-reviewed, and documented set of programs. The initial task resulted in a system design requirements report, based on input from potential Hanford users, providing general descriptions of the calculations that the final programs must perform. The recommendations of that report formed the basis for the remainder of the tasks, defining the elements that determined the equation formulation and parameter selection tasks (Napier, 1988a). The appendix to that document provides a discussion of SQA issues, including responsible organizations. Napier (1995) provides a similar discussion for GENII 2.0 and states the code was developed in a similar manner. The identified user groups are EPA analysts and contractors.

Criterion 3.5 — Napier (1988b, 1988c) discuss acceptance criteria and testing for GENII 1.485. The GENII 2.0 documentation does not specifically address acceptance criteria but implies their existence by referring to code testing.

4.3.2 Sources and Method of Review

All of the documentation listed in Table 1-2 has been reviewed with attention to "Requirements," except for Item 12 (see Appendix B).

4.3.3 Software Quality-Related Issues or Concerns

The only SQA concern for GENII 2.0 was the lack of specific acceptance criteria. There are no similar concerns for GENII 1.485.

4.3.4 Other Areas for Improvement

No other areas of improvement were noted.

4.3.5 Recommendations

Recommendations related to this topical area are provide in Table 4.3-2.

Table 4.3-2 — Recommendations for Requirements Phase Topic

Recom-mendation Number	Relates to Table 4.3-1 Criterion Number(s)	Recommendation	Est. FTE to Complete	Est. Calendar Duration
3.1	3.5	Develop and document acceptance criteria for GENII 2.0.	One FTE week	One month

4.4 Topical Area 4 Assessment: Design Phase

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

4.4.1 Criterion Specification and Result

This topical area is "graded" for GENII 1.485 and "required" for GENII 2.0. 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.	Yes for both	1.485: Napier (1988a) provides System Requirements as well as software design. 2.0: Napier (2002b) is the System Design Document

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
4.2	Code developer(s) prescribed and documented the design activities to the level of detail necessary to permit the design process to be carried out and to permit verification that the design met requirements.	Yes for both	1.485: Napier (1988a) provides System Requirements as well as software design activities. 2.0: Napier (2002b) is the System Design Document. Pseudo-code listings provided.
4.3	Design presents and documents specification of interfaces, overall structure (control and data flow) and the reduction of the overall structure into physical solutions (algorithms, equations, control logic, and data structures).	Yes for both	1.485: Napier (1988a, b, c) document overall structure, interfaces, control and data flow, and physical solutions. 2.0: Napier (1995, 2002b) document overall structure, interfaces, control and data flow, and physical solutions. Pseudo-code listings are provided. For both, diagrams show the flow of data and logic.
4.4	Design presents and documents 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 for both	1.485: Napier (1988ab,c) show that the overall system design accounted for hardware and software interfaces and limitations, including the O/S. 2.0: Napier (1,995, 2002b) provides similar features.
4.5	Design presents and documents that as an integral part of software design, problems are mitigated. These potential problems include external and internal abnormal conditions and events that can affect the computer program.	Yes for 1.485. Partial for 2.0.	1.485: Napier (1988b) provides error-reporting forms to testers and users so that errors can be fixed and users informed. 2.0: the error-reporting is less formal
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.	Yes for both	1.485: Napier (1988a) describes major components of design 2.0: Napier (2002b) is the System Design Document. Pseudo-code listings provided.

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
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.	Yes for both	1.485: Napier (1988a) provides the theoretical basis, control logic and flow, data flow and structure, mathematical models, process flow and structure, physical models, and coupling between structure and standards. 2.0: Napier (2002b) provides similar information. Pseudo-code listings are provided.
4.8	A Software Design Document, or equivalent, is available and contains a description of the allowable or prescribed ranges for inputs and outputs.	Yes for both	1.485: Napier (1988a) discusses ranges of input variables and error message generated when out of range. 2.0: Napier (2002b) provides similar information.
4.9	A Software Design Document, or equivalent, is available and contains the design described in a manner that can be translated into code.	Yes for both	1.485: Napier (1988a) and its appendix provide enough detail that the design can be translated into code 2.0: Napier (2002b) provides similar information. Pseudo-code listings are provided.
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.	Yes for both	1.485: Napier (1988a, b, c) discuss testing and the H/W and S/W configurations 2.0: Napier (1995, 2002b) provides similar information.
4.11	The organization responsible for the design identified and documented the particular verification methods to be used and assured that an Independent Review was performed and documented. This review evaluated the technical adequacy of the design approach; assured internal completeness, consistency, clarity, and correctness of the software design; and verified that the software design is traceable to the requirements.	Yes for 1.485. No for 2.0	1.485: Napier (1988a, b, c) states that the code has been thoroughly tested and verified by independent reviewers according to NQA-1 standards. 2.0: Because this code has not been completed in all its aspects, the final testing has not yet been done.

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
4.12	The organization responsible for the design assured that the test results adequately demonstrated the requirements were met.	Yes for 1.485. No for 2.0	1.485: Napier (1988a, b, c) states that the code has been thoroughly tested and verified by independent reviewers according to NQA-1 standards. 2.0: Because this code has not been completed in all its aspects, the final testing has not yet been done.
4.13	The Independent Review was performed by competent individual(s) other than those who developed and documented the original design, but who may have been from the same organization.	Yes for 1.485. No for 2.0	1.485: Napier (1988a, b, c) states that the code has been thoroughly tested and verified by independent reviewers according to NQA-1 standards. This includes review by competent, independent individuals. 2.0: Because this code has not been completed in all its aspects, the final testing has not yet been done.
4.14	The results of the Independent Review are documented with the identification of the verifier indicated.	Yes for 1.485. No for 2.0	1.485: The independent reviewers are identified by name in the Acknowledgements section of Napier (1988a,b) 2.0: Because this code has not been completed in all its aspects, the final testing has not yet been done.
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.	Yes for both	During code development, extensive manual calculations were made (and archived) to verify proper performance of the code. See final paragraph of this subsection.

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
4.16	Software design documentation was completed prior to finalizing the Independent Review.	Yes for both	1.485: Napier (1988a) states that the code has been thoroughly tested and verified by independent reviewers according to NQA-1 standards. This includes completion of S/W design prior to finalizing independent review. 2.0: Napier (2002b), the design document, has been completed. The final independent review has not yet occurred.
4.17	The extent of the Independent Review and the methods chosen are shown to be a function of the following: The importance to safety The complexity of the software The degree of standardization The similarity with previously proven software	Yes for both	These issues are decided by the independent reviewers, not the code developers. Therefore, they are not specifically addressed in the documentation of either version of GENII.

Additional Detail

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

Criterion 4.1 — The Napier (1988a) appendix, *Hanford Environmental Dosimetry Upgrade Project (HEDUP) Task 02 - System Design Requirements*, is the complete SQA requirements document for GENII 1.485. It includes the following:

1. General computational requirements
2. Computational facilities, hardware, and databases
3. Code language
4. Coding Standard and coding standard tools
5. Input parameters and format:
   - Release category and source term
   - Scenarios
   - Meteorology
   - Environmental transport
   - Exposure pathways
6. Dosimetry specifications
7. Risk assessment calculations
8. Integration of separate codes
9. Customized pathway requirements
10. Specialized scenario requirements
11. Output format

12. Graphics
13. Documentation and instructions
14. Error messages
15. Updates and revisions
16. Security
17. Quality assurance
18. Training

Napier (2002b) is the System Design Document for GENII 2.0. It defines details of the overall structure of the software, the major software components, their data file interfaces, and specific mathematical models to be used. The design represents a translation of the requirements (Napier, 1995) into a description of the software structure, software components, interfaces, and necessary data. The design focuses on the major components and data communication links that are key to the implementation of the software within the operating framework.

Criterion 4.5 — The error reporting forms for GENII 1.485 (see Figure 4-1) provided a formal method of problem mitigation. A similar methodology does not exist for GENII 2.0.

Criterion 4.10 — The hardware requirements for GENII 1.485 are an IBM PC/AT or compatible computer, an 80287 math coprocessor, 640 KB of random access memory, a minimum of 5 MB on-line disk storage, and operating under DOS 3.1 or later (Napier, 1988b). Hardware requirements for GENII 2.0 are Windows® 95, 98, NT, or 2000[2], using Pentium processors, and disk storage in excess of 60 MB. FRAMES and GENII make use of the memory swapping capabilities of Windows, so the programs should run on any Windows-compatible computer. However, they will generally run fastest on machines with 256Mbytes of memory or more (Napier, 2002a). GENII 2.0 will not run in the DOS environment.

Criterion 4.13 — GENII 1.485 has already been thoroughly reviewed and tested and there are no plans to pursue these issues again. GENII 2.0 has been reviewed at PNNL and several EPA clients, and it went through an advisory review with the EPA Science Advisory Board. This board suggested some additional capabilities that have not yet been implemented. The code author developed the code as general-purpose software and "importance to safety" was not an issue in its development. Standardization was an important consideration and was a direct response to the issue of testability and complexity of the older version. GENII 2.0 is very similar to 1.485 but it is not the same and is intended for a different set of users.

In summary, the GENII 1.485 User's Guide (Napier, 1988b), p 5.1, states: "The design process consisted of developing and internally testing software, developing test cases, and documenting software in accordance with the design input. The GENII package has been extensively tested and verified by hand, using the hand calculation worksheets of (the Code Maintenance Manual) and benchmarked against similar Hanford environmental dosimetry programs. A 10-volume set of test documentation is available for review from the authors upon request. The design process concluded with analysis of the final design by means of a Final Internal Development Review (FIDR). Two external peer reviews were held, as described in (the Conceptual Representation volume); these constitute the FIDR for the GENII package."

[2] The documentation from which this sentence was extracted (Napier, 2002a) was written before the advent of Windows XP. Experience shows that GENII 2.0 also runs under Windows XP.

4.4.2 Sources and Method of Review

All of the documentation listed in Table 1-2 has been reviewed with attention to "Design," except for Item 12 (see Appendix B), and several e-mail communications with the code developer (Bruce Napier) have helped to clarify issues.

4.4.3 Software Quality-Related Issues or Concerns

There are no additional SQA related issues or concerns in "Design."

4.4.4 Other Areas for Improvement

No other areas of improvement have been identified.

4.4.5 Recommendations

Recommendations related to this topical area are provided in Table 4.4-2.

Table 4.4-2 — Recommendations for Design Phase Topic

Recom-mendation Number	Relates to Table 4.4-1 Criterion Number(s)	Recommendation	Est. FTE to Complete	Est. Calendar Duration
4.1	4.5	See recommendation 2.1 on criterion 2.6.		
4.2	4.11, 4.12, 4.13, 4.14	When GENII 2.0 is complete, a comprehensive independent review must be documented to cover all aspects of these items	Two FTE months	Four months

Additional Detail

No additional detail is needed on the above recommendations.

4.5 Topical Area 5 Assessment: Implementation Phase

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

4.5.1 Criterion Specification and Result

This topical area is "graded" for GENII 1.485 and "required" for GENII 2.0. 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 for 1.485. Partial for 2.0	1.485: Napier (1988c) is the code maintenance manual, containing listings of all source code. Napier (1988b) is the user's manual. 2.0: Napier (2002a) is the user's guide. Program listings are not yet published.
5.2	Implemented software was analyzed to identify and correct errors.	Yes for 1.485. Partial for 2.0.	1.485: An error reporting and corrective action process was used during development. 2.0: Used an informal error reporting process.
5.3	The source code finalized during verification (this phase) was placed under configuration control.	Yes for 1.485. No for 2.0.	1.485: Configuration control was in place during code development. Current configuration control is provided through RSICC, the distributor of the code, who will not release revised code unless tested and verified. 2.0: Code is not yet finalized.
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 for both	Although the documentation reviewed (Table 1-2) does not specifically address the items provided to the testers, the code author affirms that these items were given to them.

Additional Detail

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

Criterion 5.1 — GENII 2.0 has not been finalized. Code listings should become available after completion and final testing of code.

Criterion 5.2 — See recommendation 2.1 (on Criterion 2.6) for a discussion of this.

Criterion 5.3 — The appendix to Napier (1988a), the system design document, states: "Configuration control shall be a feature of the software to protect the basic code from unauthorized changes. A control mechanism with sign-off procedures shall be implemented to protect the software from unauthorized modifications. Needed changes shall be validated before modification are permitted." Bruce Napier is the current custodian of GENII 1.485 although at times past others had been assigned this duty. The code is distributed through RSICC at Oak Ridge, TN. Together, they provide the current configuration control.

Criterion 5.4 — The code author (Bruce Napier) states (e-mail in Appendix A): "The test cases were generally designed to meet the needs of certain types of calculation, and were done first on the computer (using the code and documentation to run) and then again on the GENII-specific

hand calculation worksheets. The criteria were that the numbers had to match to two significant figures (which is all that the GENII code transfers internally at certain steps)."

4.5.2 Sources and Method of Review

E-mails with the code author addressed some of these issues. In addition, all of the documentation listed in Table 1-2 was reviewed with attention to "Implementation," except for Item 12 (see Appendix B).

4.5.3 Software Quality-Related Issues or Concerns

There are no other SQA-related issues or concerns in "Implementation Phase."

4.5.4 Other Areas for Improvement

No other areas for improvement have been identified.

4.5.5 Recommendations

Recommendations related to this topical area are provide in Table 4.5-2.

Table 4.5-2 — Recommendations for Implementation Phase Topic

Recom-mendation Number	Relates to Table 4.5-1 Criterion Number(s)	Recommendation	Est. FTE to Complete	Est. Calendar Duration
5.1	5.1	Make GENII 2.0 code listings available upon completion and final testing of code.	One FTE week	One month

4.6 Topical Area 6 Assessment: Testing Phase

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

4.6.1 Criterion Specification and Result

This topical area is "required" for both GENII 1.485 and 2.0. 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 for 1.485. No for 2.0.	1.485: Code was validated by being thoroughly tested (Napier, 1988a, 1988b). 2.0: Code not yet completed, so testing is not complete.
6.2	Testing demonstrated the capability of the software to produce valid results for test cases encompassing the range of permitted usage defined by the program documentation. Such activities provide evidence to ensure that the software adequately and correctly performed all intended functions and does not perform adverse unintended functions.	Yes for 1.485. No for 2.0.	1.485: Code was thoroughly tested (Napier, 1988a, 1988b). 2.0: Code not yet completed, so testing is not complete.
6.3	Testing demonstrated that the computer program properly handles abnormal conditions and events as well as credible failures appropriate warning or error messages are provided to the user when the code is used improperly (e.g., an input is specified outside acceptable range).	Yes for 1.485. No for 2.0.	1.485: Code was thoroughly tested (Napier, 1988a, 1988b). 2.0: Code not yet completed, so testing is not complete.
6.4	Test Phase documentation includes test procedures or plans and the results of the execution of test cases. The test results documentation demonstrates successful completion of all test cases or the resolution of unsuccessful test cases and provides direct traceability between the test results and specified software requirements.	Yes for 1.485. No for 2.0.	1.485: Code was thoroughly tested (Napier, 1988a, 1988b). 2.0: Code not yet completed, so testing is not complete.

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
6.5	Test procedures or plans specify the following, as applicable: (1) Required tests and test sequence, (2) Required range of input parameters (3) Identification of the stages at which testing is required, (4) Requirements for testing logic branches, (5) Requirements for hardware integration, (6) Anticipated output values (7) Acceptance criteria, (8) Reports, records, standard formatting, and conventions, (9) Identification of operating environment, support software, software tools or system software, hardware operating system(s) and/or limitations.	Yes for 1.485. No for 2.0.	1.485: Code was thoroughly tested (Napier, 1988a, 1988b). 2.0: Code not yet completed, so testing is not complete.

Additional Detail

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

.

Criteria 6.1 – 6.5 — Napier (1988b) states that there is a ten-volume set of test documentation available for inspection by interested parties. These documents are not included in those reviewed here, as they are at the offices at PNNL. The GENII 2.0 User's Guide (Napier, 2002a), in reference to Version 1.485, states: "GENII Version 1 has been included in the International Atomic Energy Agency's VAMP project (VAlidation of Model Predictions - an acronym for the Coordinated Research Program on Validation of Models for the Transfer of Radionuclides in Terrestrial, Urban and Aquatic Environments), an international effort to compare environmental radionuclide transport models with measured environmental data. Results for test scenario CB (based on environmental measurements following the Chernobyl accident) indicated that dose estimates from GENII were comparable to, although slightly higher than, those of other participating models, which is consistent with its primary function as a prospective analysis tool. The models included in the code have been validated to various degrees by additional studies, however these have not been compared directly to output from the code."

4.6.2 Sources and Method of Review

All of the documentation listed in Table 1-2 has been reviewed with attention to "Testing Phase," except for Item 12 (see Appendix B).

4.6.3 Software Quality-Related Issues or Concerns

There are no other SQA-related issues or concerns in "Testing Phase."

4.6.4 Other Areas for Improvement

No other areas of improvement in the "Testing Phase" have been identified.

4.6.5 Recommendations

Recommendations related to this topical area are provide in Table 4.6-2.

Table 4.6-2 — Recommendations for Testing Phase Topic

Recom-mendation Number	Relates to Table 4.6-1 Criterion Number(s)	Recommendation	Est. FTE to Complete	Est. Calendar Duration
6.1	All	Document all testing of GENII 2.0.	Three FTE months	Six months

4.7 Topical Area 7 Assessment: User Instructions

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

4.7.1 Criterion Specification and Result

This topical area is "required" for both GENII 1.485 and 2.0. Table 4.7-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
7.1	A description of the model is documented and made available to users.	Yes for both	1.485: Napier, 1988a. 2.0: Napier, 2002b.
7.2	User's manual or guide describes software and hardware limitations and identifies/includes approved operating systems (for cases where source code is provided, applicable compilers should be noted).	Yes for both	1.485: Napier, 1988b. 2.0: Napier, 2002a. Lahey Fortran-77 or F-99 compiler used. Source code in: 1.485: Napier, 1988c. 2.0: Not provided.

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
7.3	User's manual or guide includes description of the user's interaction with the software.	Yes for both	1.485: Napier, 1988b. 2.0: Napier, 2002a and 2003.
7.4	User's manual or guide includes a description of any required training necessary to use the software.	Yes for 1.485. No for 2.0.	1.485: A required training course is described in the system requirements document, not the user's manual. 2.0: Training is available (e.g., at EFCOG meetings) but it is not described in the User's Manual.
7.5	User's manual or guide includes input and output specifications.	Yes for both	1.485: Napier, 1988b. 2.0: Napier, 2002a.
7.6	User's manual or guide includes a description of user messages initiated because of improper input and how the user can respond.	Yes for both	1.485: Napier, 1988b. 2.0: Napier, 2002a.
7.7	User's manual or guide includes information for obtaining user and maintenance support.	Yes for 1.485. Partial for 2.0.	1.485: Readme.93 file on Distribution Disk 03. 2.0: Napier, 2002a.

Additional Detail

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

Criterion 7.2 — Both versions of GENII were written and compiled using the Lahey Fortran (F-77 or F-99) software, except for the user interface of GENII 1.485 (Apprentice), which was written using Microsoft QuickBasic. Source code for GENII 1.485 is given in Volume 3 of PNL-6584, *Code Maintenance Manual* (Napier, 1988c). It is also can be found on Distribution Disk02 by double clicking on SOURCE.EXE, which will unpack all the routines, both those in Fortran and those in QuickBasic. Source code is not provided for GENII 2.0.

Criterion 7.4 — The appendix to Napier (1988a), the system requirements document, p A.15, states: "A short training program shall be developed at the completion of the code to instruct potential users on the execution of the code. A detailed stepwise instruction manual shall also be prepared. Training should consist of class sessions and hand-out instructions, with opportunity for hands-on testing of the code." This training was provided on GENII 1.485 after it was released but such training is no longer available. Training for GENII 2.0 has been available at annual EFCOG meetings but there is no guarantee this will continue. Training would be useful for GENII (either version). The intuitive nature of the user interface and the documentation (e.g., Napier, 1988b, 2002a, 2003) is helpful but not enough for a first-time user.

Criterion 7.6 — In GENII 1.485, user input is primarily through the Apprentice program, which prompts the user for input and requires incorrect or incompatible entries to be corrected. Appendix B of the GENII 1.485 User's Manual (Napier, 1988b) gives an extensive discussion of error handling within GENII, not just that of Apprentice. For GENII 2.0, the FRAMES user interface provides error messages when input is incomplete, out of bounds, or conflicting.

However, the current version has bugs. For example, it is possible to be trapped in an unending loop of error messages.

Criterion 7.7 — The GENII 1.485 User's Manual gives the names of the authors of GENII but not the contact information. The primary contact person is the lead author of the code, Bruce Napier (509-375-3896). In addition, RSICC has provided a "Readme" file with the name and telephone number of a very knowledgeable user of the code (Paul D. Rittman - 509-376-8715), who can also be contacted in case of problems. For GENII 2.0, the FRAMES Constituent Database user interface gives the contact information for the lead author of GENII (Bruce Napier).

4.7.2 Sources and Method of Review

The user's manual for GENII 1.485, *GENII – The Hanford Environmental Radiation Dosimetry Software System. Volume 2: User's Manual* (Napier, 1988b), was reviewed for this Gap Analysis. Section 2 of that document gives the code overview, including user interaction levels and data file descriptions. Section 3 gives specific user instructions for both user interaction levels 0 and 1. Section 4 discusses system requirements and Section 5 discusses quality assurance topics. Appendix A gives an input/output example and Appendix B gives an extensive discussion of error messages. A revision to some of the data files for GENII 1.485 was issued in 1993 and another in 1996, but these did not change the code or its usage.

The User's Guide for GENII 2.0, *GENII Version 2 User's Guide* (Napier, 2002a) and *Getting Started with GENII Version* 2 (Napier, 2003) were reviewed for this Gap Analysis. The User's Guide provides details on all the options available in GENII 2.0, whereas the Getting Started document provides an introduction useful for evaluating simple, but typical, scenarios.

Correspondence (e-mails and telephone conversations) with an expert user of GENII 2.0 and with Bruce Napier has also been reviewed. A condensed version of them is included as Appendix A of this document. The expert user of GENII 2.0 was identified by Bruce Napier as William Joyce[3].

4.7.3 Software Quality-Related Issues or Concerns

An item not discussed in the documentation is memory management. GENII 1.485 was developed in the DOS environment and was expected to be run in that environment. Experience shows that it can be run in a DOS window in the Windows environment[4] but problems may be encountered when runs are made on Windows-XP based computer. This may stem from the fact that memory management is different between DOS and Windows. GENII 1.485 should be recompiled with a Windows-XP compatible FORTRAN compiler and verified to run properly.

The bug in error handling of GENII 2.0 (see Criterion 7.6) needs to be fixed.

---

[3] Mr. Joyce is a Senior Safety Engineer with ATL International, Corp., 20010 Century Blvd, Suite 500, Germantown, MD 20874.

[4] The Radiation Safety Information Computational Center (RSICC) at Oak Ridge verified the performance of GENII 1.485 on a 486 PC under the MS DOS 6.2 and Windows 95 operating systems. Testing conducted during the preparation of this Gap Analysis shows that GENII 1.485 also can be executed in Windows 98SE and but problems were encountered when run under Windows-XP.

4.7.4 Other Areas for Improvement

The GENII 2.0 user guidance (Napier, 2002b, 2003) doesn't always match the operations the user needs to perform. For example, in a number of cases, the instructions say to right-click a button whereas the correct procedure is a left-click. In addition, some of the screens the user sees are not in the same order given in the guidance.

GENII 1.485 can determine 95th percentile consequences in only one direction (sector) at a time. It would be very helpful to the analyst for GENII 1.485 to automatically determine the 95th percentile consequences in every sector at the site boundary and other user-selected distance (such as 100 m). This can be done now only by setting up multiple runs of GENII 1.485. GENII 2.0 cannot determine 95th percentile consequences except perhaps in a manner involving a random sampling of the weather and compiling statistics that would yield 95th percentile values. However, this has not yet been tested.

4.7.5 Recommendations

Recommendations related to this topical area are provide in Table 4.7-2.

Table 4.7-2 — Recommendations for User Instructions Topic

Recom-mendation Number	Relates to Table 4.7-1 Criterion Number(s)	Recommendation	Est. FTE to Complete	Est. Calendar Duration
7.1	Criterion 7.2	Recompile GENII 1.485 with a Windows-XP compatible compiler and verify that it runs correctly in a Windows environment.	One FTE week	Two weeks
7.2	Criterion 7.5	Correct the user guidance for GENII 2.0.	One FTE week	Two weeks
7.3	Criterion 7.6	The error message-handling problem needs to be fixed.	One FTE week	Two weeks

Additional Detail

Recommendation 7.1 – The estimate of one FTE week is for recompiling and the comparison testing, which would consist of running the same scenarios side by side on DOS-based and Window-based computers. The current version of GENII 1.485 should not be run on Windows-XP based computers.

4.8 Topical Area 8 Assessment: Acceptance Test

This area corresponds to the requirement entitled *Acceptance Test* Table 3-3 of the DOE SQA plan (DOE 2003e). During this phase of the software development, the software becomes part of a system incorporating applicable software components, hardware, and data, and is accepted for use.

During development of the software, the developing organization is responsible for documenting its procedures and acceptance tests it uses. Once the software is released, user organizations need a test

protocol to determine if the software is correctly installed. Implementation for this type of acceptance testing is the responsibility of the user organization.

Criterion Specification and Result

This topical area is "required" for both GENII 1.485 and 2.0. 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 for 1.485. No for 2.0.	1.485: Napier (1988b) states that the code was tested on PCs from many manufacturers. 2.0: Acceptance testing is not yet complete but Napier (2002a) states but the test plan has been developed and testing underway.
8.2	To the extent applicable to the developer, acceptance testing was performed prior to approval of the computer program for use.	Yes for 1.485. No for 2.0.	1.485: The code delivered to RSICC for distribution had been tested prior to release. 2.0: Acceptance testing is not yet complete.
8.3	The acceptance testing comprehensively evaluates software performance against specified software requirements. To the extent applicable to the developer, software validation was performed to ensure that the installed software product satisfies the specified software requirements.	Yes for 1.485. No for 2.0.	Both codes were developed under NQA-1 guidelines. This includes testing against software requirements. 1.485: Acceptance testing complete and code in use. 2.0: Acceptance testing is not yet complete.
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 for 1.485. No for 2.0.	1.485: Extensive test documentation is available on all aspects of code development. 2.0: Acceptance testing is not yet complete.

Additional Detail

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

> Criterion 8.1 — The GENII 1.485 User's Manual (Napier, 1988b), p 4.1, states: "Portions of the GENII Software Package have been tested on a number of IBM-PC/AT compatible machines. Versions of GENII have been established on microcomputers manufactured by GRID, NEC, Hewlett-Packard, and IBM. The IBM machines have included the new PS/2 System 50 and System 80. No machine-based incompatibilities have been found." The GENII 2.0 User Guide (Napier, 2002a), p 6, states: "A comprehensive test plan has been developed and testing is underway."

Criterion 8.2 — The preface to the RSICC distribution package of GENII 1.485 states that the authors of the code affirm that the code was tested prior to submission to RSICC for distribution to users.

Criterion 8.3 — The GENII 2.0 User Guide (Napier, 2002a), pp 5-6 states: "Both GENII versions were developed under QA plans based on the American National Standards Institute (ANSI) standard NQA-1 as implemented in the PNNL Quality Assurance Manual. All steps of the code development have been documented and tested, and hand calculations have verified the code's implementation of major transport and exposure pathways for a subset of the radionuclide library. A collection of hand calculations and other verification activities is available. A comprehensive test plan has been developed and testing is underway." The latter sentence refers to GENII 2.0, not 1.485.

Criterion 8.4 — Napier (1988b) states that there is a ten-volume set of test documentation available for inspection by interested parties.

4.8.1 Sources and Method of Review

All of the documentation listed in Table 1-2 has been reviewed with attention to "Acceptance Test," except for Item 12 (see Appendix B). The list in Appendix B includes a summary of developer/user testing and peer review of GENII for which documentation is available.

4.8.2 Software Quality-Related Issues or Concerns

There are no other SQA-related issues or concerns in "Acceptance Test."

4.8.3 Other Areas for Improvement

No other areas of improvement have been identified.

4.8.4 Recommendations

Recommendations related to this topical area are provide in Table 4.8-2.

Table 4.8-2 — Recommendations for Acceptance Test Topic

Recom-mendation Number	Relates to Table 4.8-1 Criterion Number(s)	Recommendation	Est. FTE to Complete	Est. Calendar Duration
8.1	All	Complete the documentation of acceptance testing for GENII 2.0	Two FTE months	Four months

4.9 Topical Area 9 Assessment: Configuration Control

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

4.9.1 Criterion Specification and Result

This topical area is "required" for both GENII 1.485 and 2.0. Table 4.9-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
9.1	For the developers, the methods used to control, uniquely identify, describe, and document the configuration of each version or update of a computer program (for example, source, object, and 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.	Yes for both	1.485: Configuration control followed PNO-MA-70, the PNL version of the NQA-1 Quality Assurance Manual that existed during development. In addition, a series of "software change packets" have been maintained. 2.0: Formal procedures for configuration control follow the current PNNL "Software Based Management System" (SBMS). Notebooks and backups are also used for this purpose. (See Appendix A.)
9.2	Implementing procedures meet applicable criteria for configuration identification, change control, and configuration status accounting.	Yes for both	See the comments above, for Criterion 9.1.

Additional Detail

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

> Criteria 9.1 and 9.2 — Configuration control followed/follows procedures formalized in SQA methods used at PNL/PNNL during the development of each version of GENII. These

procedures have evolved over the years, and thus, the procedures used for Version 2.0 are not identical to those used for Version 1.485. The author of the code(s) has kept informal notebooks and copies of earlier versions.

4.9.2 Sources and Method of Review

All of the documentation listed in Table 1-2 has been reviewed with attention to "Configuration Control," except for Item 12 (see Appendix B), as well as e-mails with the code developer.

4.9.3 Software Quality-Related Issues or Concerns

There are no SQA-related issues or concerns in "Configuration Control."

4.9.4 Other Areas for Improvement

No additional areas of improvement in "Configuration Control" have been identified.

4.9.5 Recommendations

There are no recommendations related to 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 the DOE SQA plan (DOE 2003e).

4.10.1 Criterion Specification and Result

This topical area is "graded" for both GENII 1.485 and 2.0. Table 4.10-1 lists the subset of criteria reviewed for this topical area and summarizes the findings.

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
10.1	The developing organization's problem reporting and corrective action process addresses the appropriate requirements of its corrective action system and is documented in implementing procedures.	Yes for 1.485. No for 2.0	Napier (1988b) discusses how to report errors and request upgrades. An informal method is used for GENII 2.0. See criterion 2.6.

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

Criterion Number	Criterion Specification	Compliant	Summary Remarks
10.2	The process for evaluating, and documenting whether a reported problem is an error is documented and implemented.	No for both	Not specifically discussed in the documentation reviewed. However, the SQA procedures followed during development (see criterion 9.1) do require problem reporting and documenting.
10.3	The process for disposition of the problem reports, including notification to the originator of the results of the evaluation, is documented and implemented.	No for both	Not specifically discussed in the documentation reviewed. However, the SQA procedures followed during development (see Criterion 9.1) do require proper disposition of problem reports.
10.4	A documented process provides guidance on determining how identified errors relate to appropriate software engineering elements and is implemented.	No for both	Not discussed in the documentation reviewed.
10.5	The process is documented and implemented for determining how an error impacts past and present use of the computer program.	No for both	Not discussed in the documentation reviewed.
10.6	The process is documented and implemented for determining how an error and resulting corrective action impacts previous development activities.	No for both	Not discussed in the documentation reviewed.
10.7	The process is documented and implemented describing how the users are notified of an identified error, its impact; and how to avoid the error, pending implementation of corrective actions.	No for both	Not discussed in the documentation reviewed.

4.10.2 Sources and Method of Review

All of the documentation listed in Table 1-2 has been reviewed with attention to "Error Impact," except for Item 12 (see Appendix B).

4.10.3 Software Quality-Related Issues or Concerns

For users of GENII 2.0 within PNNL, the existing Standards Based Management System (SBMS) process can be followed. There would be no software quality-related issues or concerns for these users. However, for users outside of PNNL, the process of error notification and corrective action needs to be formalized and documented so that users know how to report errors, how PNNL will respond, how PNNL will notify other users of the problem, and how too avoid the problem.

4.10.4 Other Areas for Improvement

No other areas of improvement are noted.

4.10.5 Recommendations

Recommendations related to this topical area are provide in Table 4.10-2.

Table 4.10-2 — Recommendations for Error Impact Topic

Recom-mendation Number	Relates to Table 4.13-1 Criterion Number(s)	Recommendation	Est. FTE to Complete	Est. Calendar Duration
10.1	All	A formal error reporting and corrective action process needs to be implemented for GENII 1.485 and GENII 2.0 for users outside of PNNL.	One FTE month	Two months

4.11 Training Program Assessment

No regularly scheduled GENII training program is conducted. Training materials for Version 1.485 of GENII are still available, but there have been no requests made to the author (Bruce Napier) to use these for several years.

There have been discussions with the EPA about training on Version 2, and the author has given some Version 2.0 training at recent EPA NESHAPS meetings (held annually). Future training may be provided to the NRC headquarters staff. However, the latter is still in the planning stage.

The last known training to DOE safety analysis community occurred during the 2000 Energy Facility Contractors Group (EFCOG) Safety Analysis Working Group Workshop (April 2000). It is recommended that this forum be explored to provide DOE users with a regular opportunity for GENII training.

5.0 Conclusion

The gap analysis for Version 1.485 and 2.0 of the GENII 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), nine requirements are met at an acceptable level for GENII 1.485, (items 1-9). Improvement actions are recommended for GENII 1.485 to fully meet the requirement for *Error Impact* (item 10). For GENII 2.0, of the ten primary SQA requirements for existing software at the Level B classification (important for safety analysis but whose output is not applied without further review), two requirements are met at an acceptable level, i.e., *Software Classification* (1) and *Configuration Control* (9). Improvement actions are recommended for GENII 2.0 to fully meet the requirement for five that are partially met, i.e., *SQA Procedures and Plans* (2), *Requirements Phase* (3), *Design Phase* (4), *Implementation Phase* (5), and *User Instructions* (7) and for the remaining three, *Testing Phase* (6), *Acceptance Test* (8), and *Error Impact* (10). This evaluation outcome is deemed acceptable because: (1) GENII 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 GENII is limited to those analytic applications for which the software is intended.

It was determined that GENII 1.485 code does meet its intended function for use in supporting documented safety analysis, providing it is not used on a Windows-XP based computer. It was determined that GENII 2.0 will not meet its intended function. Therefore, only GENII 1.485 can be recommended for DSA use at this time. As with all safety-related software, users should be aware of current limitations and capabilities of GENII for supporting safety analysis. Informed use of the software can be assisted by the current set of GENII reports (refer to Table 1-2), and the code guidance report for DOE safety analysts, *GENII Computer Code Application Guidance for Documented Safety Analysis*, (DOE, 2004). Furthermore, while SQA improvement actions are recommended for GENII, no evidence has been found of programming, logic, or other types of software errors in GENII 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 GENII software improvement actions be taken, especially:

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

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

Recommendations are given for each of the topical areas in Section 4.0. It is estimated that nearly ten full-time equivalent (FTE) months would be required to perform all SQA upgrade tasks covered in Section 4.0 for GENII 2.0. Because GENII 1.485 has been in use for many years and the code author does not intend to make any further modifications, no similar estimates need be made. The error-reporting estimate for GENII 2.0 may be applied to GENII 1.485. In order to use GENII 1.485 in all Windows environments, it will be necessary to recompile the code using a Windows-XP compatible compiler. A side-by-side testing on DOS-based and Windows-based computers would then follow this. The GENII 1.485 documentation would not need to be changed if the results were the same but documentation of the results should be included with the RSICC distribution package for GENII 1.485. The recompiled version would have to be given a new number, such as 1.486.

Training opportunities exist for both versions of GENII, but these are not routinely offered. It is recommended that user training for safety analysis applications be conducted formally on at a minimum, an annual basis. Prerequisites for, and core knowledge needed by, the user prior to initiating GENII applications should be documented by the code developer.

While completion of the GENII 2.0 development is encouraged, current DOE DSA support should be through the earlier code version, GENII 1.485. Use of Windows-XP based computers should be avoided for GENII 1.485 until such time that a Windows-XP based version is available.

6.0 Acronyms and Definitions

ACRONYMS:

ALOHA	Areal Locations of Hazardous Atmospheres (designated toolbox software)
ANS	American Nuclear Society
ANSI	American National Standards Institute
ASME	American Society of Mechanical Engineers
CD	Compliance Decision
CFAST	Consolidated Fire and Smoke Transport Model (designated toolbox software)
CFR	Code of Federal Regulations
DNFSB	Defense Nuclear Facilities Safety Board
DoD	Department of Defense
DOE	Department of Energy
DSA	Documented Safety Analysis
EFCOG	Energy Facility Contractors Group
EH	DOE Office of Environment, Safety and Health
EIS	Environmental Impact Statement
EPA	Environmental Protection Agency
EPIcode	Emergency Prediction Information code (designated toolbox software)
FTE	Full-time equivalent
GENII	Generalized Environmental Radiation Dosimetry Software System - Hanford Dosimetry System (Generation II) (designated toolbox software)
IEEE	Institute of Electrical and Electronics Engineers
IP	Implementation Plan
ISO	International Standards Organization
MACCS2	MELCOR Accident Consequence Code System 2 (designated toolbox software)
MELCOR	Methods for Estimation of Leakages and Consequences of Releases (designated toolbox software)
NESHAPS	National Emission Standards for Hazardous Air Pollutants
NNSA	National Nuclear Security Administration
NRC	Nuclear Regulatory Commission
QAP	Quality Assurance Program (alternatively, Plan)
RSICC	Radiation Safety Information Computational Center
SNL	Sandia National Laboratories
SQA	Software Quality Assurance
V&V	Verification and Validation

DEFINITIONS

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Safety Software — Includes both safety system software, and safety analysis and design software. [DOE O 414.1B]

Safety System Software — Computer software and firmware that performs a safety system function as part of a structure, system, or component (SSC) that has been functionally classified as Safety Class (SC) or Safety Significant (SS). This also includes computer software such as human-machine interface software, network interface software, programmable logic controller (PLC) programming language software, and safety management databases that are not part of an SSC but whose operation or malfunction can directly affect SS and SC SSC function. [DOE O 414.1B]

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

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

Software design
Implementation
Test, and sometimes:
Installation. [NQA-1]

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

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

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

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

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

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

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

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

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

7.0 References

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

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

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

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

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

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

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

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

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

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

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

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

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

Napier, 1988a, B. A. Napier, R. A. Peloquin, D. L. Strenge, and J. V. Ramsdell, *GENII – The Hanford Environmental Radiation Dosimetry Software System. Volume 1: Conceptual Representation.* PNL-6584, Pacific Northwest Laboratories, Richland, WA, (December 1988).

Napier, 1988b, B. A. Napier, R. A. Peloquin, D. L. Strenge, and J. V. Ramsdell, *GENII – The Hanford Environmental Radiation Dosimetry Software System. Volume 2: User's Manual*, PNL-6584, Pacific Northwest Laboratories, Richland, WA, (November 1988).

Napier, 1988c, B. A. Napier, R. A. Peloquin, D. L. Strenge, and J. V. Ramsdell, *GENII – The Hanford Environmental Radiation Dosimetry Software System. Volume 3: Code Maintenance Manual*, PNL-6584, Pacific Northwest Laboratories, Richland, WA, (September 1988).

Napier, 1995, B. A. Napier, J. V. Ramsdell, and D. L. Strenge. *Software Requirements Specifications for Hanford Environmental Dosimetry Coordination Project*, May 1995 Draft Report, prepared for review by the EPA Office of Radiation and Indoor Air.

Napier, 1999a, B. A. Napier, *GENII Version 2 Example Calculation Descriptions*, Prepared for U.S. Environmental Protection Agency, January 1999.

Napier, 1999b, B. A. Napier and L. Staven, *GENII Version 2 Training Power Point Slides*, Presented at the Safety Analysis Workshop of the annual meeting of the Energy Facility Contractors Group (EFCOG), June 1999.

Napier, 2002a, B. A. Napier, *GENII Version 2 User's Guide*, Prepared for U.S. Environmental Protection Agency, September 2002.

Napier, 2002b, B. A. Napier, D.L. Strenge, J. V. Ramsdell, Jr., P. W. Eslinger, and C. Fosmire, *GENII Version 2 Software Design Document*, Prepared for U.S. Environmental Protection Agency, November 2002.

Napier, 2003, B. A. Napier, *Getting Started with GENII Version 2*, Prepared for U.S. Environmental Protection Agency, February 2003.

Napier, 2003, B. A. Napier, Communication with K.R. O'Kula.

Appendices

Appendix	Subject
A	COMMUNICATIONS WITH OTHERS
B	GENII BENCHMARKING AND V&V

APPENDIX A.— COMMUNICATIONS WITH OTHERS

Appendix A is provided to supplement and support information found in GENII documentation. Bruce Napier is the code developer and William Joyce is an EPA consultant that has been reviewing GENII 2.0.

1) Differences between Versions 1.485 and 2.0

Response from Napier:
Version 2 is much different than 1.485:

- It uses hourly meteorology, not joint frequency data.
- It is set up for the acute release met model to start at a defined date and time. However, the FRAMES system has a stochastic processor that wraps around all the GENII modules and allows variation in all the input parameters.
- It can be run a few thousand times, varying the start time. This has the effect of building the entire output dose distribution, not just the 95th percentile meteorology.
- Seasonal model taken out only, uses the Fall model.

2) Topics regarding SQA

Response from Napier:

GENII 1.485

This version was developed under the earliest NQA-1 standards (1986 version):

SQA Plan — Exists, out of date. Refers to PNNL manual no longer available. Napier has key chapters though.

Software Requirements Document — Exists, but the one developed was very short, and not nearly as detailed as may now be desired.

Software Design Document —Believe the GENII PNL-6854 Volume 1 report covers this.

Test Case Description and Report — Ran all modifications against a series of regression tests with known answers. Have an extensive series of documented hand calculation worksheets that give "the right answer." Some documentation is available not in reports.

Software Configuration and Control Document— Hard copies of all the versions from 1.350 (the point at which things were stable) through 1.485, including the software change packets exist. RSICC does code distributions.

Error Notification and Corrective Action Report — Not done now except in extraordinary circumstances.

User's Manual, and other relevant documentation (model description, weekly or monthly reports to code sponsor, etc.). — Believe GENII PNL-6854 Volume 2 report covers this.

No official changes to the code since 1990. Because of compiler and other issues, changes may be difficult now.

GENII V2

GENII Version 2 keeps the name, and a few of the basic algorithms. Almost everything else is new.

The formal QA is weaker than for 1.485.

SQA Plan — Exists, but is very brief.

Software Requirements Document — Exists, reasonably detailed and complete.

Software Design Document —GENII Version 2 Software Design Document available.

Test Case Description and Report — None exists.

Software Configuration and Control Document —Informal notebooks and backups exist.

Error Notification and Corrective Action Report — Limited beta testing is done.

User's Manual, and other relevant documentation (model description, weekly or monthly reports to code sponsor, etc.). — GENII Version 2 Users Guide is available, plus the "Getting Started with GENII" exists.

Was reviewed by the EPA Science Advisory Board (who have a report).

3) Verification Documentation -- copy of the software available to testers, test case description available, associated criteria available, software requirements traceability.

Response from Napier:

The test cases were generally designed to meet the needs of certain types of calculation, and were done first on the computer (using the code and documentation to run) and then again on the GENII-specific hand calculation worksheets. The criteria were that the numbers had to match to 2 significant figures (which is all that the GENII code transfers internally at certain steps).

- Verifiers had the software.
- Verifiers had the documentation. The GENII documentation, PNL-6584 Volume 1 contains the Design Requirements as an appendix. Requirements are traceable.
- Verifiers had test case descriptions (or wrote their own).
- Verifiers had criteria.

4) General code use and information

Response from William Joyce:

- GENII 2.0 can't give 95th percentile consequences
- The ten receptor locations in GENII 2.0 are each forced to be at the nearest grid points, which may not be where the user wants them
- GENII 2.0 is meant for EPA NESHAPS *(was not developed specifically for DSA applications)*
- GENII 1.485 was developed in a DOS environment and therefore had to address the memory limit of <640 KB. The Windows memory management system is different and there is a potential that this may lead to problems.

APPENDIX B. — GENII BENCHMARKING AND V&V

(List provided by Bruce Napier)

Publications on GENII Verification and Validation

Johnson, K.A., and M.J. Sowa. 1997. Benchmarking the GENII and RESRAD Computer Codes, Oregon State University Radiation Center, Corvallis, Oregon.

International Atomic Energy Agency. 1995. Validation of Models using Chernobyl Fallout Data from the Central Bohemia Region of the Czech Republic: Scenario CB, IAEA-TECDOC-795, First Report of the VAMP Multiple Pathways Assessment Working Group, International Atomic Energy Agency, Vienna, Austria.

Maheras, S.J. 1995. *GENII Version 1.485* (Software Review), Health Physics, 68, pp. 119-121.

Rittmann, P.D. 1995. Benchmarking of Computer Codes (GENII, PATHRAE, RESRAD) Using Hand Calculations, Westinghouse Hanford Company, Richland, Washington.

Maheras, S.J., P.D. Ritter, P.R. Leonard, and R. Moore. *Benchmarking of the CAP-88 and GENII Computer Codes using 1990 and 1991 Monitored Atmospheric Releases from the Idaho National Engineering Laboratory*, Health Physics, 67, pp. 509-517.

Faillace, E.R., J.J. Cheng, and C. Yu. 1994. RESRAD Benchmarking Against Six Radiation Exposure Pathway Models, ANL/EAD/TM-24, Argonne National Laboratory, Argonne, Illinois.

Preece, A.B. 1993. Use of the GENII Computer Code in a Low-Level Radioactive Waste Disposal Facility Performance Assessment Methodology. Master's Thesis, University of Texas, Austin, Texas.

Seitz, R.R., J.R. Cook, M.I. Wood, P.D. Rittmann, B.A. Napier, and D.W. Wood. 1992. Comparison of Computer Codes and Inputs Used at DOE Sites to Model Intrusion Scenarios, PNL-SA-20502, Pacific Northwest Laboratory. Presented at Waste Management '92, Tucson, Arizona, March 1-5, 1992

Kozak, M.W., M.S.Y. Chu, and P.A. Mattingly. 1990. A Performance Assessment Methodology for Low-Level Waste Facilities, NUREG/CR-5532, Sandia National Laboratories, Albuquerque, New Mexico.

Kozak, M.W., M.S.Y. Chu, P.A. Mattingly, J.D. Johnson, and J.T. McCord. 1990. Background Information for the Development of a Low-Level Waste Performance Assessment Methodology: Identification and Recommendation of Computer Codes, NUREG/CR-5453, Volume 5, Sandia National Laboratories, Albuquerque, New Mexico.

Jaquish, R. E., and B. A. Napier. 1987. "A Comparison of Environmental Radionuclide Concentrations Calculated by a Mathematical Model with Measured Concentrations." PNL-SA-14720. In Proceedings of ANS Topical Conference on Population Exposure from the Nuclear Fuel Cycle. Oak Ridge, Tennessee.

Aaberg, R. L., and B. A. Napier. 1985. Hanford Dose Overview Program: Comparison of AIRDOS-EPA and Hanford Site Dose Codes, PNL-5633, Pacific Northwest Laboratory, Richland, Washington.

Additional GENII Benchmarking and Comparisons

Stull, E. 1990. Comparison of GENII and RSAC-4 for use in the New Production Reactor EIS program.

Ikenberry, T.A. 1990. Demonstration of acceptable accuracy and reproducibility of the HUDU atmospheric dispersion and radiation dose program (benchmark against GENII).

Peterson, V., R. Patlovany, and G. Ennis. 1992. Comparison of MACCS and GENII, EG&G Rocky Flats, Boulder, Colorado.

Sartori, E., A. Curti, L. Riposi, and G. Graziani. 1992. Comparison of GENII and VADOSCA Computer Codes. Nuclear Energy Agency, Commission of the European Communities.

Abbott, M. 1993. MACCS2 benchmarking against GENII, Idaho National Engineering Laboratory, Idaho Falls, Idaho.

Aaberg, R.L. 1993. Comparison of GENII and RSAC-5 for the Tank Waste Remediation System (TWRS). Pacific Northwest National Laboratory, Richland, Washington.

Morris, J., and C. Williams. 1994. Workshop to discuss the use of environmental transport and fate models in the Environmental Restoration part of the DOE Programmatic Environmental Impact Statement (PEIS).

Peer Review of Multimedia Models. 1994. Comparison of MEPAS, MMSOIL, RESRAD, and GENII. M. Small (Carnegie-Mellon), D. Back (HydroGeologic), R. Charbeneau (U. Texas), C. Chein (duPont), Y. Cohen (U. California), T. Gallagher (HydroQual), M. Kavanaugh (Montgomery Watson), J. Mauro (SC&A), E. Makhlouf (Montgomery Watson), and B. Weiss (U. Rochester).

Seitz, R., P.D. Rittmann, J. Cook, and M. Wood. 1994. Comparison of PATHRAE and GENII, DOE Performance Assessment Task Team.

Summary of Developer/User Testing and Peer Review of GENII for which Documentation is Available

Baker, D. 1987. Review of HEDUP documentation and QA.

Carter, M.(Georgia Institute of Technology), K. Eckerman (Oak Ridge National Laboratory), J. Johnson (Chalk River Laboratory). 1987. External peer review panel.
R.Gray. 1988. Extramural panel review of GENII.

Napier, B.A. 1990. GENII "Conversion Testing, Verification, and Validation of Software" plan listing 42 tests performed as of 2/7/1989.

Rhoads, K. 1990. Review of acute dispersion calculation GENII Version 1.449.

Winter, R., G. Anast, H. Avci, M. Biggerston, D. Smith. 1990. Informal review of GENII verification and validation. Argonne National Laboratory/ DOE-HQ.

Nelson, I.C., L.H. Sawyer, T.A. Ikenberry. 1990. Hand Calculations performed on GENII to support NPR-EIS program.

Sawyer, L.H., T.A. Ikenberry. 1991. Hand calculations performed to support acute models in GENII.

Cammann, J. and P.D. Rittmann. 1990. Revisions to GENII dose increment libraries.

Peloquin, R.A., 1994. GENII Hand Calculation Worksheets, version of February 2, 1994.