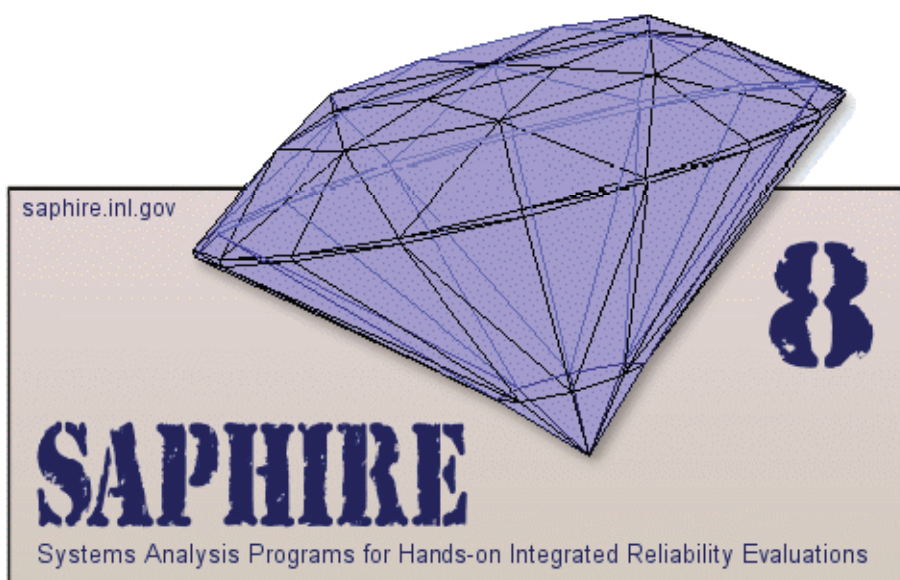


Independent Verification and Validation of SAPHIRE 8 Software Requirements

March 2010



The INL is a U.S. Department of Energy National Laboratory
operated by Battelle Energy Alliance

Independent Verification and Validation of SAPHIRE 8 Software Requirements

March 2010

**Idaho National Laboratory
Idaho Falls, Idaho 83415**

<http://www.inl.gov>

**Prepared for the
U.S. Nuclear Regulatory Commission
Washington, DC 20555
Project Number N6423**

Table of Contents

1.0	Executive Summary	1
2.0	Background Information	2
3.0	Summary of Findings.....	3
3.1	NUREG/BR-0167 Findings	3
3.2	SAPHIRE Version 8 SVVP – Volume I Findings	4
4.0	IV&V Evaluation Checklist.....	6

1.0 Executive Summary

The purpose of the Independent Verification and Validation (IV&V) role in the evaluation of the SAPHIRE 8 requirements definition is to assess the activities that results in the specification, documentation, and review of the requirements that the software product must satisfy, including functionality, performance, design constraints, attributes and external interfaces. The IV&V team began this endeavor after the software engineering and software development of SAPHIRE 8 had already been in production. IV&V reviewed the requirements specified in the NRC Form 189s to verify these requirements were included in the SAPHIRE Version 8 Software Verification and Validation Plan (SVVP) Volume I (INL/EXT-05-00821).

The requirements for IV&V review were extracted primarily from the NUREG/BR-0167 Software Quality Assurance Program and Guidelines, but also included an examination of best software engineering methods provided in the IEEE Standard for Software Verification and Validation. IV&V developed a checklist that mapped these requirements with these standards which was used in the evaluation. The evaluation criteria and the results of the assessment are identified in section 4 of this document.

Traceability of requirements is the greatest of these concerns. Requirements traceability is essential to all software development activities. Without a well documented Requirements Traceability Matrix (RTM), design components may be overlooked, and test cases missed.

For IV&V to properly evaluate the RTM to assess the mapping of the test cases to design components and to requirements as documented in the SAPHIRE Version 8 SVVP, IV&V had to obtain requirements from the Statement of Work documents (Form 189s) and develop the RTM. This action could place IV&V's "independence" role into question. The intent of IV&V in developing the RTM is strictly for use in evaluation and not intended for use by the development team. However, the RTM will be included as documentary evidence in the IV&V report provided to the sponsor and the INL Project Manager.

Per the requirements and document outline provided in the SAPHIRE 8 Software Independent Verification and Validation Plan (INL/EXT-09-15649), this report and all subsequent reports will be included as attachments and/or background evidence of the evaluation as well as the results of the assessment.

2.0 Background Information

NUREG/BR-0167, Software Quality Assurance Program and Guidelines, requires the development of Software Requirements Documentation that specifies the requirements that the software to be developed/maintained must meet. An item can be called a software requirement only if its achievement can be verified and validated. It is important that each software requirement be traceable throughout the stages of the software life cycle.

This report provides an evaluation of the Software Requirements Documentation. The Software Requirements Documentation is intended to provide the specification, documentation, and review of the requirements to meet the contractual commitments prepared by the sponsor; the Nuclear Regulatory Commission.

Independent Verification and Validation (IV&V) evaluates and assesses the processes and products developed during each phase of the Software Development Life Cycle (SDLC). The SAPHIRE 8 development team is implementing a “spiral” rapid application approach to the product development. One of the roles that IV&V performs, regardless of the development methodology, is to analyze products developed throughout the development process. The intent is to provide a level of confidence to the sponsor that the quality of the software product and supporting documentation is built into the software, not tested in. Evaluating the supporting documentation for each product is one aspect of providing this level of confidence.

IV&V supports and is complementary to the Quality Assurance, Project Management, and product development activities. To achieve this support, IV&V must also evaluate the processes identified in the documentation to ensure that the development team is implementing the processes and methodology that ensures a high-level software product.

Due to the spiral approach implemented for the software development, it is expected that the Software Requirements Documentation will evolve as the SAPHIRE 8 product matures. Therefore, IV&V will evaluate each iteration of the Software Requirements Documentation.

To provide direction in the evaluation process, IV&V has developed a checklist to support the requirements for the SDLC. The Project Plan requirements used for the analysis of the Software Requirements Documentation is contained in a checklist that is included in the SAPHIRE 8 Software Independent Verification and Validation Plan (INL/EXT-09-15649). The evaluation criteria for the Software Requirements Documentation have been extracted from the checklist contained in the “IV&V Plan” and included in section 4 of this report. A summary of the findings is provided in section 3.

3.0 Summary of Findings

An Independent Verification and Validation evaluation of the Software Verification and Validation Plan – Volume I, section 2 Software Requirements and section 3 Interface Requirements Specification, Document ID: INL/EXT-05-00821 for SAPHIRE 8 was performed using the checklist contained in section 4.0 of this document. The checklist was extracted from the SAPHIRE 8 Software Independent Verification and Validation Plan Document ID: INL/EXT-09-15649. Section 3.1 refers to the specific parts of the NUREG/BR-0167 Software Quality Assurance Program and Guidelines requirements the SAPHIRE 8 Software Verification and Validation Plan – Volume I sections 2 and 3 failed to satisfy. Of the 21 criteria listed in the checklist contained in section 4.0 of this document 6 failed. In order for these criteria to pass, the requirements in section 2 Software Requirements and section 3 Interface Requirements Specification within the Software Verification and Validation Plan – Volume I need to be uniquely identified. Section 3.2 of the Summary of Findings lists minor corrections for the Software Verification and Validation Plan – Volume I sections 2 Software Requirements and 3 Interface Requirements Specification.

3.1 NUREG/BR-0167 Findings

Section 4.3 Software Requirements Documentation requires:

Software requirements documentation specifies the requirements that the software to be developed/maintained must meet. Include in this documentation the following, as applicable:

1. Functionality – the functions that the software is to perform.
2. Performance – the time-related requirements of software operation such as speed, response time, etc.
3. Design constraints imposed on implementation activities – any elements that will restrict design options (e.g., specifying the hardware platform or the programming language).
4. Attributes – characteristics of the software, its acceptance, or use (e.g., portability, acceptance criteria, access control, availability, maintainability, etc.).
5. External interfaces – interactions with people, hardware, and other software.

An item can be called a software requirement only if its achievement can be verified and validated. It is important that each software requirement be traceable throughout the stages of the software life cycle.

Criteria 1 – Sub-sections within section 2 Software Requirements and section 3 Interface Requirements Specification contain paragraphs specifying multiple requirements and single sentences containing a single requirement. In order for the requirements to be testable and traceable through the software life cycle the individual requirements need to be uniquely identified. In order to uniquely identify the requirement, identify each requirement as a “functional (e.g., FR-01)”, “performance (e.g., PR-01)”, “design constraint (e.g., DCR-01)”, “attribute (e.g., AR-01)”, or “external interface (e.g., IR-01)” requirement. This unique identifier then becomes the requirement id listed in the Requirements Traceability Matrix, which is then

traceable to a design component id and test case id listed in the Requirements Traceability Matrix. Refer to section 4 IV&V Evaluation Checklist criteria 1 for more information.

Criteria 9 – Sub-sections within section 2 Software Requirements and section 3 Interface Requirements Specification contain paragraphs specifying multiple requirements and single sentences containing a single requirement. In order for the requirements to be testable and traceable through the software life cycle the individual requirements need to be uniquely identified.

Criteria 10 – Sub-sections within section 2 Software Requirements and section 3 Interface Requirements Specification contain paragraphs specifying multiple requirements and single sentences containing a single requirement. In order for the requirements to be testable and traceable through the software life cycle the individual requirements need to be uniquely identified. In order to uniquely identify the requirement, identify each requirement as a “functional (e.g., FR-01)”, “performance (e.g., PR-01)”, “design constraint (e.g., DCR-01)”, “attribute (e.g., AR-01)”, or “external interface (e.g., IR-01)” requirement. This unique identifier then becomes the requirement id listed in the Requirements Traceability Matrix, which is then traceable to a design component id and test case id listed in the Requirements Traceability Matrix.

Criteria 11 – Sub-sections within section 2 Software Requirements and section 3 Interface Requirements Specification contain paragraphs specifying multiple requirements and single sentences containing a single requirement. In order for the requirements to be testable and traceable through the software life cycle the individual requirements need to be uniquely identified. In order to uniquely identify the requirement, identify each requirement as a “functional (e.g., FR-01)”, “performance (e.g., PR-01)”, “design constraint (e.g., DCR-01)”, “attribute (e.g., AR-01)”, or “external interface (e.g., IR-01)” requirement. This unique identifier then becomes the requirement id listed in the Requirements Traceability Matrix, which is then traceable to a design component id and test case id listed in the Requirements Traceability Matrix. Refer to section 4 IV&V Evaluation Checklist criteria 11 for more information.

Criteria 13 – The requirements as listed in the SAPHIRE Version 8 Software Verification and Validation Plan – Volume II (INL/EXT-05-00821) appendix D Requirements Traceability Matrix (RTM) are R1 thru R74. In order for the requirements to be testable and traceable through the software life cycle the individual requirements need to be uniquely identified. In order to uniquely identify the requirement, identify each requirement as a “functional (e.g., FR-01)”, “performance (e.g., PR-01)”, “design constraint (e.g., DCR-01)”, “attribute (e.g., AR-01)”, or “external interface (e.g., IR-01)” requirement. This unique identifier then becomes the requirement id listed in the Requirements Traceability Matrix, which is then traceable to a design component id and test case id listed in the Requirements Traceability Matrix. Refer to section 4 IV&V Evaluation Checklist criteria 13 for more information.

The RTM does not list design components that map to requirements. The test cases as listed in the RTM are incomplete (“NA”, “None”, “To be determined”).

Criteria 14 – Some of the requirements as listed in the RTM do not map to a test case (“NA” and “None”).

3.2 SAPHIRE Version 8 SVVP – Volume I Findings

1. Page 1, Section 1 Introduction and Overview, second paragraph.

When referring to SAPHIRE Version 8, “8” is used and “8.0” is used.
IV&V suggests using “SAPHIRE Version 8” to be consistent with the rest of the document.

2. Page 7, Section 1.4 SAPHIRE Features, Table 1.4 Salient features of SAPHIRE and relevant version numbers.
Under “L Basic-Event Calculations”, the Description of Feature “Common-cause plug-in modules” does not list a “Item” or “X” under the columns Version 7.x and Version 8.x.
3. Page 12, Section 2.1 Graphical User Interface Requirements.
The first sentence references “NRC Form 189, Revision 10”.
IV&V suggests supplying the JCN to be consistent with the previous section 2 Software Requirements which lists the JCN when referencing the NRC Form 189s.
4. Page 12, Section 2.1.3 Project Integration and Modification, following the third paragraph.
Is “Project Directory Structure” a section heading?
5. Page 12, Section 2.1.3 Project Integration and Modification, the first sentence following “Project Directory Structure”.
“A subdirectory structure will exist under a project's folder is used to store information in an organized and standardized format.”
IV&V suggests removing “is” from the sentence.
6. Page 19, Section 2.5.1 Standard Analysis.
This section does not contain any information.
7. Page 21, Section 2.5.8 Rules Editor.
IV&V suggests following the sentence *“Rules are used in conjunction with the analysis techniques describe above to customize the process and provide specific results.”*
Add “..., SAPHIRE will:” to be consistent with the previous sections containing bullet items describing the SAPHIRE functionality.
8. Page xiv, ACRONYMS.
The following acronyms in the acronym list are not used in the document:
BWR, FEP, IPE, IRRAS, NPP, PWR, SARA.

The following acronyms are used in the document, but are not listed in the acronym list:
DOE, EE, IRS, LHS, MCS, NASA, PEP, PRRA, QA, QL, SDS, SSC, SQAP, SRS, SVVP,
TM, UID.

The following acronym is used in the document with two definitions:
PM.

4.0 IV&V Evaluation Checklist

SOFTWARE REQUIREMENTS It is assumed that shall and will are requirement identifications whereas should and would are “statements of fact” and not considered “testable” requirements.		
Criteria: 1	Does the Requirements Document identify requirements that are uniquely identified, testable, and traceable through the software life cycle? NUREG/BR-0167 Section 4.3	
Pass		Comments
Fail	X	<p>Section 2 Software Requirements provides requirements obtained from NRC Forms 189s Y6394 Revision 11, N6203 Revision 3, N6423 Revision 7, and Y6394 Revision 10. Section 3 Interface Requirements Specification provides requirements for the types of interfaces SAPHIRE 8 will provide. Sub-sections within section 2 Software Requirements and section 3 Interface Requirements Specification contain paragraphs specifying multiple requirements and single sentences containing a single requirement. Requirements are identifiable by the use of “will” and “shall” within the paragraphs and single sentences. In order for the requirements to be testable and traceable through the software life cycle the individual requirements need to be uniquely identified. In order to uniquely identify the requirement, identify each requirement as a “functional (e.g., FR-01)”, “performance (e.g., PR-01)”, “design constraint (e.g., DCR-01)”, “attribute (e.g., AR-01)”, or “external interface (e.g., IR-01)” requirement. This unique identifier then becomes the requirement id listed in the Requirements Traceability Matrix, which is then traceable to a design component id and test case id listed in the Requirements Traceability Matrix.</p> <p>For example: Section 2.1.1 Access to Top Level Objects contains a single requirement. This requirement could be considered a “Functional” requirement since it describes a function that the software will perform or this requirement could be considered an “External Interface” requirement since the requirement will interact with people. If uniquely identifying the requirement as a external interface requirement, the requirement could be uniquely identified as:</p> <p><i>“IR-01 – The GUI will provide access to the following top-level objects: Fault tree objects for creation, modification, and running analysis scenarios; Event tree objects for creation, modification, and running analysis scenarios; Event objects and basic events for creation, modification, and running analysis scenarios; Graphical editor tool-set for creating model representations; Workspace area for project environment settings, security, and controls.”</i></p> <p>Section 2.1.1 Project Controls contains multiple requirements. These requirements could be considered “Functional” requirements since they describe functions that the software will perform. If uniquely identifying the requirements as functional requirements, the requirements could be uniquely identified as:</p> <p><i>“FR-01 - The GUI will provide an enhanced environment for the management of a multiuser/multiproject environment. This will include object level security and sharing of objects between projects.”</i> <i>“FR-02 – The GUI will provide support for a standard directory structure.”</i> <i>“FR-03 – The GUI will provide support for search capabilities.”</i></p>
N/A		

Criteria: 2	Does the Requirements Document address the functions that the software is to perform and only what is to be performed? NUREG/BR-0167 Section 4.3.1, Software Engineering Practices	
Pass	X	Comments
Fail		Sub-sections within section 2 Software Requirements and section 3 Interface Requirements Specification contain paragraphs specifying multiple requirements and single sentences containing a single requirement. Requirements identified by the use of “will” and “shall” within the paragraphs and single sentences address the functions that the software is to perform.
N/A		
Criteria: 3	Does the Requirements Document address time-related requirements of software operation such as speed, response time, and/or other performance requirements? NUREG/BR-0167 Section 4.3.2	
Pass	X	Comments
Fail		Section 2.1 Graphical User Interface Requirements specifies “ <i>The purpose of the upgraded interface is to create a series of Web-based, HTML-like screens with dropdown menus and form fields to facilitate user interaction. It will combine elements of both the old SAPHIRE and GEM tools in an attempt to improve usability by giving users access to most of the commonly used options and features of the tool set with a GUI.</i> ”
N/A		
		Section 2.1.2 Project Controls specifies “ <i>The GUI will provide an enhanced environment for the management of a multiuser/multiproject environment.</i> ”
		Section 2.1.3 Project Integration and Modification specifies “ <i>Facilitate an increase in project quality by enforcing object restriction rights and tracking modifications.</i> ”
		Section 2.5 Core Analysis Requirements specifies “ <i>SAPHIRE will be able to perform (both quantification and reporting of results) standard risk/reliability analyses. SAPHIRE will be able to perform analyses for the Significance Determination Process (SDP). SAPHIRE will be able to perform analyses for Events and Condition Assessment (ECA) (formally known as Accident Sequence Precursor, ASP, analysis).</i> ”
		Section 2.8 External Events Requirements specifies “ <i>The application will extend the analysis capability to allow for external events modeling.</i> ”
		Requirements identified by the use of “will” and “shall” within the paragraphs and single sentences address performance requirements of the software.
Criteria: 4	Does the Requirements Document address constraints imposed on implementation activities, including but not limited to hardware platform and programming language? NUREG/BR-0167 Section 4.3.3	
Pass	X	Comments
Fail		Section 2.2 Application Program Interface (API) Requirements, section 2.4.1 Model Creation and Maintenance, section 2.4.6 Sequence Object, section 2.5 Core Analysis Requirements, section 2.5.7 User-Defined Model Type, section 3.2 Application Program Interface Requirements and section 3.10 Operating System Interface Requirements address design constraint requirements of the software.
N/A		
Criteria: 5	Does the Requirements Document address attributes of the software, such as portability, access controls, property of an object, element, or file? NUREG/BR-0167 Section 4.3.4 – Best Practices	
Pass	X	Comments
Fail		Sub-sections within section 2 Software Requirements and section 3 Interface Requirements Specification address attributes of the software.
N/A		

Criteria: 6		Does the Requirements Document identify external interfaces – interactions/communications with people, hardware, and other software? NOTE: Interfaces may be identified in a separate document, e.g., an Interface Requirements Specification. NUREG/BR-0167 Section 4.3.5
Pass	X	Comments
Fail		Section 3 Interface Requirements Specification addresses the interface requirements of the software.
N/A		
Criteria: 7		Does the Requirements Document identify internal interfaces – interactions/communications which exist between separate software components and provide a programmatic mechanism by which these components can communicate? NOTE: Interfaces may be identified in a separate document, e.g., an Interface Requirements Specification. NUREG/BR-0167 Section 2.2, Section 3.2.2.1 – Section 3.2.4.1 -Software Best Practices
Pass	X	Comments
Fail		Section 3 Interface Requirements Specification addresses the interface requirements of the software.
N/A		
Criteria: 8		Does the Requirements Document identify assumptions, constraints, or dependencies that the requirements are based upon? NUREG/BR-0167 Section 4.3, Software Best Practices
Pass	X	Comments
Fail		Section 2 Software Requirements and section 2.1 Graphical User Interface Requirements refer to the Nuclear Regulatory Commission (NRC) Form 189s that the requirements are based upon.
N/A		
Criteria: 9		Is each requirement uniquely identified and requirements baseline under CM control? NUREG/BR-0167 Section 6.2
Pass		Comments
Fail	X	Sub-sections within section 2 Software Requirements and section 3 Interface Requirements Specification contain paragraphs specifying multiple requirements and single sentences containing a single requirement. Requirements are identifiable by the use of “will” and “shall” within the paragraphs and single sentences. In order for the requirements to be testable and traceable through the software life cycle the individual requirements need to be uniquely identified. Refer to Criteria 1.
N/A		
		The requirements as identified in the SAPHIRE Version 8 Software Verification and Validation Plan – Volume 1 (INL/EXT-05-00821) are baselined using the revision control system (RCS) described in the Software Configuration Management Plan (INL/EXT-09-16696).
Criteria: 10		Are the requirements verifiable (clarity increases verifiability)? NOTE: A requirement is verifiable if some method can be devised for objectively demonstrating that the software implements it. NUREG/BR-0167 Section 3.2.1.5
Pass		Comments
Fail	X	Sub-sections within section 2 Software Requirements and section 3 Interface Requirements Specification contain paragraphs specifying multiple requirements and single sentences containing a single requirement. Requirements are identifiable by the use of “will” and “shall” within the paragraphs and single sentences. In order for the requirements to be testable and traceable through the software life cycle the individual requirements need to be uniquely identified. In order to uniquely identify the requirement, identify each requirement as a “functional (e.g., FR-01)”, “performance (e.g., PR-01)”, “design constraint (e.g., DCR-01)”, “attribute (e.g., AR-01)”, or “external interface (e.g., IR-01)” requirement. This unique identifier then becomes the requirement id listed in the Requirements Traceability Matrix, which is then traceable to a design component id and test case id listed in the Requirements Traceability Matrix.
N/A		

Criteria: 11	Does each statement of a requirement contain one and only one requirement? Are all requirements identified uniquely and unambiguous? (Functional, Performance, Design Constraints, Attribute, Interfaces). Do requirements state WHAT and not HOW they are implemented? Note: Interface requirements may be included in the SRS if not in a separate document. NUREG/BR-0167 Section 3.2.1.5	
Pass		Comments
Fail	X	Sub-sections within section 2 Software Requirements and section 3 Interface Requirements Specification contain paragraphs specifying multiple requirements and single sentences containing a single requirement. Requirements are identifiable by the use of “will” and “shall” within the paragraphs and single sentences. In order for the requirements to be testable and traceable through the software life cycle the individual requirements need to be uniquely identified. In order to uniquely identify the requirement, identify each requirement as a “functional (e.g., FR-01)”, “performance (e.g., PR-01)”, “design constraint (e.g., DCR-01)”, “attribute (e.g., AR-01)”, or “external interface (e.g., IR-01)” requirement. This unique identifier then becomes the requirement id listed in the Requirements Traceability Matrix, which is then traceable to a design component id and test case id listed in the Requirements Traceability Matrix. Refer to Criteria 1.
N/A		
Criteria: 12	Is there a Requirements Traceability Matrix? NUREG/BR-0167 Section 3.2.1.5	
Pass	X	Comments
Fail		Refer to SAPHIRE Version 8 Software Verification and Validation Plan – Volume II (INL/EXT-05-00821) appendix D.
N/A		
Criteria: 13	Does the Requirements Traceability Matrix (RTM) provide the preliminary trace of Functional Requirements (e.g., FR-01), Performance Requirements (e.g., PR-01), Design Constraint Requirements (e.g., DCR-01), Attribute Requirements (e.g., AR-01), and Interface Requirements (e.g., IR-01) down to the unit level and do test cases map to requirements? NUREG/BR-0167 Section 3.2.1.5	
Pass		Comments
Fail	X	The requirements as listed in the SAPHIRE Version 8 Software Verification and Validation Plan – Volume II (INL/EXT-05-00821) appendix D Requirements Traceability Matrix (RTM) are R1 thru R74. In order for the requirements to be testable and traceable through the software life cycle the individual requirements need to be uniquely identified. In order to uniquely identify the requirement, identify each requirement as a “functional (e.g., FR-01)”, “performance (e.g., PR-01)”, “design constraint (e.g., DCR-01)”, “attribute (e.g., AR-01)”, or “external interface (e.g., IR-01)” requirement. This unique identifier then becomes the requirement id listed in the Requirements Traceability Matrix, which is then traceable to a design component id and test case id listed in the Requirements Traceability Matrix. Refer to Criteria 1.
N/A		The RTM does not list design components that map to requirements. The test cases as listed in the RTM are incomplete (“NA”, “None”, “To be determined”).
Criteria: 14	Are all requirements testable? (If it is not testable, then it is not a requirement) NUREG/BR-0167 Section 1.7, Table 1-1, Section 2.1, Section 2.5.2, Table 3-1, Section 3.2.2.3	
Pass		Comments
Fail	X	Some of the requirements as listed in the RTM do not map to a test case (“NA” and “None”).
N/A		
Criteria: 15	Is the RTM under Configuration Management and Change Control? NOTE: The RTM is a living document and should be <u>baselined</u> at the end of each life-cycle phase or when changes to requirements occur within a life-cycle phase after it has been baselined. NUREG/BR-0167 Table 1-1, Section 6, Section 6.2	

Pass	X	Comments
Fail		The Requirements Traceability Matrix in the SAPHIRE Version 8 Software Verification and Validation Plan – Volume II (INL/EXT-05-00821) appendix D is baselined using the revision control system (RCS) described in the Software Configuration Management Plan (INL/EXT-09-16696).
N/A		
Criteria: 16	Does the Requirements Document identify the purpose and scope? NUREG/BR-0167 Section 2.2, 4.3	
Pass	X	Comments
Fail		The required information is provided in section 1 Introduction and Overview and section 2 Software Requirements.
N/A		
Criteria: 17	Does the Requirements Document identify what the products will and will not do? Software Engineering Practices	
Pass	X	Comments
Fail		This information is provided in section 1.4 SAPHIRE Features, Table 1.4 Salient features of SAPHIRE and relevant version numbers.
N/A		
Criteria: 18	Does the Requirements Document describe the objectives and goals? NUREG/BR-0167 Section 5.2.1	
Pass	X	Comments
Fail		The required information is provided in section 1 Introduction and Overview.
N/A		
Criteria: 19	Does the Requirements Document describe any constraints on memory or other system constraints? Software Engineering Practices	
Pass	X	Comments
Fail		Section 2.2 Application Program Interface (API) Requirements, section 2.4.1 Model Creation and Maintenance, section 2.4.6 Sequence Object, section 2.5 Core Analysis Requirements, section 2.5.7 User-Defined Model Type, section 3.2 Application Program Interface Requirements and section 3.10 Operating System Interface Requirements address design constraint requirements of the software.
N/A		
Criteria: 20	Does the Requirements Document describe backup and recovery operations, if applicable? Software Engineering Practices	
Pass	X	Comments
Fail		Section 2.2 Application Program Interface (API) Requirements specifies “ <i>The API interface to the database will provide database operations to perform searches, inputs, edit, and delete data. There will be the capability to load data into the database and to extract data from the database into a flat file format, such as a spreadsheet or delimited text file</i> ”.
N/A		
Criteria: 21	Does the Requirements Document describe assumptions? (Assumptions can lead into Risks) Software Engineering Practices	
Pass	X	Comments
Fail		Assumptions are addressed in the SAPHIRE Version 8 Software Verification and Validation Plan – Volume 1 (INL/EXT-05-00821) section 4 Design Specification.
N/A		