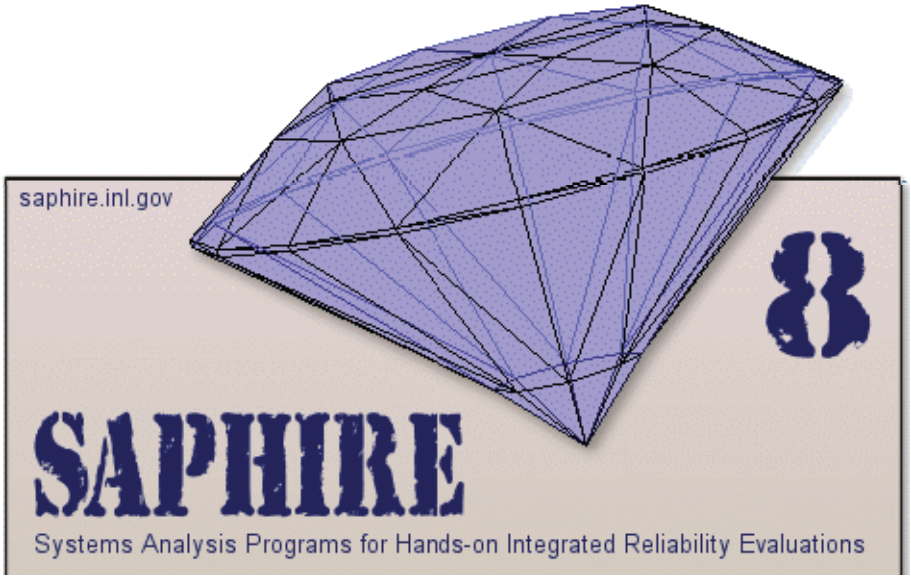
INL/EXT-09-16789

Independent Verification and Validation of SAPHIRE 8 Software Requirements

September 2009



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

Idaho National Laboratory

INL/EXT-09-16789

Independent Verification and Validation of SAPHIRE 8 Software Requirements

September 2009

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

The purpose of the Independent Verification and Validation (IV&V) role in the evaluation of the SAPHIRE 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 had already been in production. IV&V reviewed the requirements specified in the NRC Form 189s to verify these requirements were included in SAPHIRE's Software Verification and Validation Plan (SVVP).

The requirements for IV&V review were extracted primarily from the NUREG 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 5 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 SAPHIRE's Software Verification and Validation Plan, 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 IV&V Plan, 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, Revision ID: 0, Effective Date: April 1, 2009). The evaluation criteria for the Software Requirements Documentation have been extracted from the checklist contained in the "IV&V Plan" and included in section 5 of this report. A summary of the findings is provided in section 3.

3.0 Summary of Findings

The requirements are included in section 2 Software Requirements and section 3 Interface Requirements Specification within the SAPHIRE Version 8 Software Verification and Validation Plan Volume I. Section 2 Software Requirements, states "*The requirements are briefly described in NRC Form 189, Y6394, Revision 11 and NRC Form 189 N6203, Revision 3*". Section 2.1 Graphical User Interface Requirements, states "*The graphical user interface (GUI) requirements are described in NRC Form 189, Revision 10, subtasks 9.5 and 9.6*". Requirements are also found in:

- NRC Form 189, Y6394, Revision 7,
- NRC Form 189, N6423, Revision 0 and
- NRC Form 189, N6423, Revision 5.

There is no Software Requirements Documentation containing all requirements.

The requirements are not all documented or uniquely identified, but requirements that are documented are specified in paragraphs and referenced in the Requirements Traceability Matrix as section numbers within the SAPHIRE Version 8 Software Verification and Validation Plan Volume I. Sentences within paragraphs containing the words "shall" and "will" are assumed requirements. Other sentences containing the words "is", "must", "can be" and "are" were assumed requirements. Since multiple requirements are specified within paragraphs and not uniquely identified, traceability and testability of individual requirements will be difficult and violates sound software engineering practices.

3.1 NUREG/BR-0167 Findings

3.1.1 Section 2.2 Requirements Definition

Fail

The requirements definition process is the set of activities that result in the specification, documentation, and review of the requirements that the software product must satisfy, including functionality, performance, design constraints, attributes, and external interfaces.

Ensure that the documented requirements define the response of the software to anticipated classes of input data (including erroneous data) and provide the information and detail necessary to design the software (e.g., mathematical models, equations, data requirements).

Define the requirements so that they are correct, complete, verifiable, consistent, and technically feasible. Perform planning activities for verification and validation in parallel with requirements definition activities.

Because requirements inevitably change as a project evolves, manage the requirements throughout the development and maintenance efforts in accordance with well defined change control procedures.

3.1.2 Section 3.2.1 Verification and Validation Planning Activities

Fail

Planning for verification and validation takes place during the sponsor's initial planning for the project (e.g., the proposal stage) as well as during the requirements definition, design, and implementation major activities of the life cycle. Planning activities include:

1. Development or tailoring of procedures for conducting formal life cycle reviews.
2. Development or tailoring of procedures for reviewing documentation and other deliverables.
3. Development or tailoring of procedures for conducting inspections.
4. Definition of a detailed test methodology, including standards for test documentation, specifically for test plans test procedures, and test reports for both qualification and acceptance testing.
5. Preparation of a validation matrix showing the relationship of software requirements to the software architecture down to the unit level and to the tests used to verify the requirements.
6. Identifying the need for development and test tools, equipment, and data.

3.1.3 Section 3.2.2.1 Software Requirements Review

Fail

Conduct the Software Requirements Review at the end of requirements definition. The primary objective of this review is to assure that the sponsor and the developer understand and agree on the intent, completeness, verifiability (through testing or other means), consistency, and technical feasibility of the requirements. Secondary objectives are to review other documentation products available at this time, including, for example, the Software Project Plan and the overall verification and validation plan.

3.1.4 Section 3.2.4.1 Design-Driven and Requirements-Driven Testing

Pass

Requirements-driven or black-box testing is done by selecting input data and other parameters based on the software requirements and observing the outputs and reaction of the software. In addition to testing for satisfaction of requirements, some of the objectives of requirements-driven testing are to ascertain:

1. Computational correctness.
2. Proper handling of boundary conditions, including extreme inputs and conditions that cause extreme outputs.
3. State transitioning as expected.
4. Proper behavior under stress or high load.
5. Adequate error detection, handling, and recovery.

Sometimes the term "operational testing" is used. Operational testing is either the random or statistically-controlled application of the software in its actual environment or in a simulated version of the operational environment. An example of such testing is the so-called beta test use of an applications software package by individuals typical of the intended user population. In the terminology used above, such operational testing and beta testing would be qualification testing and are requirements-driven.

3.1.5 Section 4.3 Software Requirements Documentation

Fail

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.

3.1.6 Section 6.2 Baselines

Pass

Establish controlled and stable baselines for planning, managing, and building the system. Explicitly identify as project baselines software products (e.g., source code, object code, test cases) and software process specifications (e.g., standards and procedures) that are needed to establish and maintain stability of the software activities.

Establish a naming or labeling system that:

1. Uniquely identifies all project entities (e.g., documents, software elements, test cases).
2. Identifies changes by revision or version.
3. Uniquely identifies each configuration/version of revised software for use.

Establish the following baselines:

1. The project management baseline consisting of the Software Project Plan, documented standards and procedures, and up-to-date budgets and schedules.
2. The requirements baseline consisting of the software requirements documentation plus approved changes.
3. The product baseline consisting of software and documentation resulting from the qualification testing activity.
4. The operational baseline consisting of software and documentation resulting from the installation and acceptance activity that is placed into operation.

The developmental configuration is the developer's software and associated technical documentation that defines the evolving software products during development. It contains the software design and implementation products (software design documentation, code, test cases, and related information). Require the developer to apply internal configuration control procedures to the developmental configuration as it evolves.

3.2 NRC Form 189 Findings

3.2.1 NRC Form Revision 07 Job Code Y6394 Requirements

NRC Form, Revision 07 Job Code Y6394 Project Start Date 12-01-2000 Project End Date 12-30-2004 Requirements. There are no specific requirements in the SAPHIRE Version 8 Software Verification and Validation Plan section 2 Software Requirements or section 3 Interface Requirements Specification referring to these NRC requirements:

- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.2: New SAPHIRE Version, Analysis Size Limitations: This task will minimize the inherent size limitations on the number of basic events, fault trees, sequences, end states, etc., handled by SAPHIRE. This will necessitate the reworking of algorithms to ensure that the speed of accessing the data was not impacted significantly by the size increase.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: SPAR Model Support: In order to extend the functionality of the SPAR models, modifications must be made to the SAPHIRE software to accommodate improvements and increase efficiencies in the model development process.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: SPAR Model Support, Automatic SPAR Document Creation: Enhance SAPHIRE to facilitate automatic creation of an electronic version of the SPAR document via a single "report-type" option. The general format (layout, headers, sections, page numbering, etc.) of the automated report will approximately follow the published Rev. 3 SPAR documents.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: SPAR Model Support, Automatic SPAR Document Creation, Specific Activities Required To Automate The Documentation, Basic Event Notes/Comments: General textual information should be accessible for the basic events in addition to use of pre-defined "bibliography-type" centralized notes/references.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: SPAR Model Support, Automatic SPAR Document Creation, Specific Activities Required To Automate The Documentation, Basic Event Attributes: General attributes for the variety of basic event types (human reliability, common- cause, hardware failures, etc.) should be published in one (or more) tables in the SPAR document. For example, the basic events that are of common-cause type would have their textual information (name, description, notes) in addition to the common-cause calculation information (failure criteria, alpha factors, related independent events).
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: SPAR Model Support, Automatic SPAR Document Creation, Specific Activities Required To Automate The Documentation, Basic Event Attributes: The SPAR-H related basic events should include information related to the specific performance shaping factors used for the basic event, uncertainty, and any dependency information.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: SPAR Model Support, Automatic SPAR Document Creation, Specific Activities Required To Automate The Documentation, Quantification Results: Output from the baseline analysis should include sequence-level results frequencies and basic event importance measures.

- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: SPAR Model Support, Automatic SPAR Document Creation, Specific Activities Required To Automate The Documentation, Graphics: The event/fault tree sections should contain their associated graphics (logic/P&ID) in sorted and numbered alphabetic order.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: SPAR Model Support, Automatic SPAR Document Creation, Specific Activities Required To Automate The Documentation, Graphics: The page layout, including headers/footers, should be user modifiable.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: SPAR Model Support, SPAR-H Worksheet Function: Create the equivalent to the SPAR-H worksheets used in the SPAR models within the SAPHIRE database. The current SPAR models use the worksheets off-line and then the value from the worksheet is transcribed into SAPHIRE. The SPAR-H function will replace the off-line calculation and perform the quantification, including uncertainty, directly in SAPHIRE.

3.2.2 NRC Form Revision 11 Job Code Y6394 Prior Efforts

NRC Form, Revision 11 Job Code Y6394 Project Start Date 12-01-2000 Project End Date 12-30-2005 Prior Efforts. There are no specific requirements in the SAPHIRE Version 8 Software Verification and Validation Plan section 2 Software Requirements or section 3 Interface Requirements Specification referring to these Prior Efforts:

- SAPHIRE Development/Maintenance: Increased the number of sequences handled by SAPHIRE from 100,000 to up to 2,000,000.
- SAPHIRE Development/Maintenance: Added a "cut set merge" option so that cut sets produced when performing Level 2 analysis could be traced back to the Level 1 fault trees.
- SAPHIRE Development/Maintenance: Expanded the cut set "slicer" option to allow cut set lists based on select events.
- SAPHIRE Development/Maintenance: Upgraded the advanced event tree linkage rules to allow user-defined variables.
- SAPHIRE Development/Maintenance: Added modules to support the SPAR model development, including custom reports and a SPAR-H analysis form.

3.2.3 NRC Form Revision 11 Job Code Y6394 Requirements

NRC Form, Revision 11 Job Code Y6394 Project Start Date 12-01-2000 Project End Date 12-30-2005 Requirements. There are no specific requirements in the SAPHIRE Version 8 Software Verification and Validation Plan section 2 Software Requirements or section 3 Interface Requirements Specification referring to these NRC requirements:

- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.1: Design and implement reports for the SPAR Large Early Release Frequency (LERF) models.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.2: New SAPHIRE Version, Parallel Processing and Multiple Processor Routines for Complex Model

Analysis: Studies have shown that complex operations can be divided up between multiple processors and completed more quickly and efficiently. This task will enhance SAPHIRE's capability to deal with extremely large and complex models.

- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.2: New SAPHIRE Version, Risk API (Application Program Interface) Support: Provide support towards developing and maintaining the calculation modules stored in the Risk API.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, External Events Sequence Generation Automation: Design and implementation of a SAPHIRE module to automatically generate applicable accident sequence logic specific to external events (e.g., fire, flood, seismic) using a "scenario generation" table.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, External Events Sequence Generation Automation: The "scenario generation" table will consist of attributes such as a scenario name, description, frequency, affected components, event tree to use, and an end state substitution. This automatically sequence generation will be used to replace the manual construction of external event sequences and subsequent duplication of logic.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, External Events Sequence Generation Automation: SAPHIRE will use the scenario generation table to automatically generate each required sequence without any addition user input.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, Internal and External Events Solving and Reporting: SAPHIRE modifications to both streamline the steps required in solving external events models and provide reporting capabilities specific to external events analysis.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, Internal and External Events Solving and Reporting: Modify SAPHIRE so that the normally individual analysis cases stored in the "analysis type" can be evaluated in a single pass.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, Internal and External Events Solving and Reporting: All relevant (user selectable) accident sequences must be able to be evaluated using a single processing step.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, Internal and External Events Solving and Reporting: The reporting of analysis type results must also be streamlined to coincide with the analysis steps.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, Internal and External Events Solving and Reporting: Modify SAPHIRE to provide an option from the main menu to allow a user to run analysis cases such as: Solve internal and external event core damage frequency (CDF); Solve just internal event CDF; Solve just external event CDF; Solve just seismic events CDF.

- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, Internal and External Events Solving and Reporting: Modifications will need to function regardless of the "analysis type" used since external events models such as seismic may be identified only by the initiating event type rather than the "analysis type" designation.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, Internal and External Events Solving and Reporting: If the user wished to evaluate just internal events, then the external events initiators would be set to zero for the analysis and the results would be returned just for internal events.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, Internal and External Events Solving and Reporting: The designation of the initiating event type will either utilize a user-defined label (e.g., "FR" to indicate fire, "FL" to indicate flood) or one of the "category type" labels.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, Internal and External Events Solving and Reporting: In order to alert the user, the analysis screen and output report(s) must provide textual feedback information pertaining to the analysis option(s) being evaluated. The large early release frequency (LERF) model endstate is not included as an option at this time.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, Internal and External Events Solving and Reporting: Modify SAPHIRE to enhance the report output capabilities to produce analysis output tables.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, Internal and External Events Solving and Reporting: These tables should be exportable to several formats including Microsoft Word, Excel, and Corel WordPerfect.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, Internal and External Events Solving and Reporting: The information provided in these tables is to be constructed from the analysis results.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, Internal and External Events Solving and Reporting: The number of tables may be as many as 15 and the specific layout will be dictated by the NRC PM by way of examples.
- Task 4: SAPHIRE Code Updates and New Capabilities, Subtask 4.3: External Events SPAR Model Support, Internal and External Events Solving and Reporting: Potential tables include: Plant CDF by event type; Plant CDF by initiating event categories; Plant CDF by dominant sequences; Plant CDF by cut set contribution; Component importance measures (e.g., RAW, RRR).
- Task 9: User-Friendly Input/Output (I/O) Interface for Use with SPAR Models: Develop a user-friendly I/O interface for use with the Level 1, Revision 3 SPAR models when performing Significance Determination Process (SDP) Phase 2 analyses and precursor analyses in the Accident Sequence Precursor (ASP) Program.

- Task 9: User-Friendly Input/Output (I/O) Interface for Use with SPAR Models: This modification will be accomplished by altering the current version of the GEM to address the needs regarding input and output that have been expressed by key users of the SDP Phase 2 analysis worksheets [e.g., regional senior reactor analysts (SRAs), senior resident inspectors (SRIs), and resident inspectors (RIs)] and key users of the SPAR models (e.g., ASP analysts, SRAs).
- Task 11: Support and Maintenance of the COOPRA Web Site: Update the structure and user interface for accessing the on-line COOPRA databases.

3.2.4 NRC Form Revision 03 Job Code Y6203 Requirements

NRC Form, Revision 03 Job Code N6203 Project Start Date 01-26-2006 Project End Date 11-30-2009 Requirements. There are no specific requirements in the SAPHIRE Version 8 Software Verification and Validation Plan section 2 Software Requirements or section 3 Interface Requirements Specification referring to these NRC requirements:

- Task 1: SAPHIRE/Graphical Evaluation Module (GEM) Maintenance and User Support: The INL shall maintain Versions 6 and 7 of the SAPHIRE/GEM code to ensure that they remain functional and state-of-the-art.
- Task 2: SAPHIRE Code Updates and Development of New Capabilities, Subtask 2.1: Common-Cause Failure (CCF) Methodology Support: The CCF module calculations will be made consistent with the RASP guidance.
- Task 2: SAPHIRE Code Updates and Development of New Capabilities, Subtask 2.1: Common-Cause Failure (CCF) Methodology Support: These modifications will be made available both in SAPHIRE and GEM.
- Task 2: SAPHIRE Code Updates and Development of New Capabilities, Subtask 2.1: Common-Cause Failure (CCF) Methodology Support: The modifications will involve changes to (1) the SAPHIRE graphical user interface (GUI) directing how end-users and modelers interface with PRA models (e.g., SPAR models) and (2) algorithmic changes as required.
- Task 2: SAPHIRE Code Updates and Development of New Capabilities, Subtask 2.1: Common-Cause Failure (CCF) Methodology Support: Included in the modifications will be the possibility to treat CCF probability adjustments based upon failure types (e.g., independent versus dependent) and failure modes (e.g., failure-to-run versus failure-to-start).
- Task 2: SAPHIRE Code Updates and Development of New Capabilities, Subtask 2.1: Common-Cause Failure (CCF) Methodology Support: Reporting capability will be augmented, including reports outlining the algorithms to be used and basic event changes.

3.2.5 NRC Form Revision 00 Job Code Y6423 Requirements

NRC Form, Revision 00 Job Code N6423 Project Start Date 05-29-2007 Project End Date 11-30-2009 Requirements. There are no specific requirements in the SAPHIRE Version 8 Software Verification and Validation Plan section 2 Software Requirements or section 3 Interface Requirements Specification referring to these NRC requirements:

- Task 4: Level 2 and LERF Module for SAPHIRE 8: Extended Level 1 SPAR models: Flagging of events added specifically to support Level 2 accident progression; Flagging of top events added specifically to support Level 2 accident sequence logic; Support graphical (event tree) display of core damage (CD) only and/or full plant damage states (PDS); Support linking in fault trees for CD or for full PDS trees.
- Task 4: Level 2 and LERF Module for SAPHIRE 8: PDS Grouping Capabilities: Support the use of decision trees/logic for defining distinct PDS groups; Support assigning sequences to PDS groups.
- Task 4: Level 2 and LERF Module for SAPHIRE 8: Level 2 Containment Phenomena Event Tree (CPET): Support modeling Level 2 phenomena that affect accident progression using event trees and subsidiary (decomposition) event trees (DETs); Support exact quantification of CPETs via "end-to-end" sequence quantification using multiplication of sequence branch split-fraction probabilities; Support approximate quantification of CPETs by marginalizing the CPETs.
- Task 4: Level 2 and LERF Module for SAPHIRE 8: Source Term Bins: Support the use of decision trees/logic for defining distinct source term bins and for assigning containment accident progression sequences to source term bins.

3.2.6 NRC Form Revision 05 Job Code Y6423 Requirements

NRC Form, Revision 05 Job Code N6423 Project Start Date 05-29-2007 Project End Date 11-30-2011 Requirements. There are no specific requirements in the SAPHIRE Version 8 Software Verification and Validation Plan section 2 Software Requirements or section 3 Interface Requirements Specification referring to these NRC requirements:

- Task 3: SPAR Model Preparation for SDP Interface: INL shall work with the SPAR model development team to load the category text related to the SDP module functionality for the Level 1 at power SPAR models. The information to be loaded includes the component, train, system, and failure mode attributes. All 73 SPAR models are applicable to this information loading effort.
- Task 5: Common Cause Failure Methodology Support for SAPHIRE 8: INL shall port the CCF methodology capability developed for SAPHIRE 7 into SAPHIRE 8.
- Task 5: Common Cause Failure Methodology Support for SAPHIRE 8: Refine the SAPHIRE 8 module and interfaces to support CCF analysis.
- Task 7: Beta Version Updates of SAPHIRE 8 for General Release: SPAR model-related updates to SAPHIRE 8 will be made as necessary.
- Task 9.2: Support for Testing, Bug Fixes, and Code Maintenance: INL will review existing legacy code (from SAPHIRE 7) to make modifications for the purpose of making it more efficient, bug free, and maintainable.
- Task 9.2: Support for Testing, Bug Fixes, and Code Maintenance: INL will continue to update select modules that are written in the Modula-2 language by porting them into the Delphi language.

- Task 9.2: Support for Testing, Bug Fixes, and Code Maintenance: INL will continue to update and maintain the SAPHIRE Risk API by making modifications to modules in the API.
- Task 9.3 Code Improvement for Common Cause Failure (CCF) Basic Events: INL will augment the CCF module to ensure that SPAR models that use fault tree and sequence flag sets have the applicable adjustments made to any impacted CCF basic events. Currently, flag sets are processes after basic event data changes, which implies that CCF may not be modified for a specific fault tree or sequence flag set. SAPHIRE will be modified to automatically treat the CCF adjustments using the new CCF module designed into SAPHIRE 8.
- Task 12: SAPHIRE Analysis Improvements: Task 12.1: Improved Algorithms to Decrease Analysis Time: SAPHIRE 8 is designed for large, complex PRA models. However, those PRAs such as the SPAR models have shown the need for analysis improvements, both in fidelity of calculations and the improvements in analysis speed. The INL will modify critical analysis modules inside of SAPHIRE 8 in order to decrease the overall time required for analysis.
- Task 12: SAPHIRE Analysis Improvements: Task 12.1: Improved Algorithms to Decrease Analysis Time: SAPHIRE 8 is designed for large, complex PRA models. However, those PRAs such as the SPAR models have shown the need for analysis improvements, both in fidelity of calculations and the improvements in analysis speed. The INL shall identify and investigate algorithm and recovery analysis changes that have the potential for improvements in analysis time.
- Task 12: SAPHIRE Analysis Improvements: Task 12.2: Decreased Analysis Time by using Multiple Computer Processors: The INL shall modify SAPHIRE to run on multi-processor computers.
- Task 12: SAPHIRE Analysis Improvements: Task 12.2: Decreased Analysis Time by using Multiple Computer Processors: The INL shall identify and investigate changes that will be able to effectively use multi-processors to decrease overall analysis time.

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 Priority: 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	The requirements are not uniquely identified. Multiple requirements are listed in paragraphs. Testability and traceability of the requirements through the software life cycle cannot be achieved. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 1	**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		Comments
Fail	X	The requirements need to be uniquely identified as functional, performance, design constraints, attributes and external interface requirements. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 1	**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		Comments
Fail	X	The requirements need to be uniquely identified as functional, performance, design constraints, attributes and external interface requirements. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 3	**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		Comments
Fail	X	The requirements need to be uniquely identified as functional, performance, design constraints, attributes and external interface requirements. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 2	**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		Comments
Fail	X	The requirements need to be uniquely identified as functional, performance, design constraints, attributes and external interface requirements. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 2	**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		Comments
Fail	X	The requirements need to be uniquely identified as functional, performance, design constraints, attributes and external interface requirements. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 1	**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		Comments
Fail	X	The requirements need to be uniquely identified as functional, performance, design constraints, attributes and external interface requirements. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 1	**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		Comments
Fail	X	The requirements need to be uniquely identified as functional, performance, design constraints, attributes and external interface requirements. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 1	**Is each requirement uniquely identified and requirements baseline under CM control? NUREG/BR-0167 Section 6.2**	
Pass		Comments
Fail	X	The requirements need to be uniquely identified as functional, performance, design constraints, attributes and external interface requirements. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 1	**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	The requirements matrix is incomplete and does not show the relationship of software requirements to the software architecture down to the unit level and to the tests used to verify the requirements. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 2	**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	The requirements are not uniquely identified. Multiple requirements are listed in paragraphs. Testability and traceability of the requirements through the software life cycle cannot be achieved. The requirements need to be uniquely identified as functional, performance, design constraints, attributes and external interface requirements. Refer to section 3.0 Summary of Findings.
N/A		

Criteria Priority: 1	**Is there a Requirements Traceability Matrix? NUREG/BR-0167 Section 3.2.1.5**	
Pass		Comments
Fail	X	The requirements matrix is incomplete and does not show the relationship of software requirements to the software architecture down to the unit level and to the tests used to verify the requirements. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 1	**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 matrix is incomplete and does not show the relationship of software requirements to the software architecture down to the unit level and to the tests used to verify the requirements. The requirements need to be uniquely identified as functional, performance, design constraints, attributes and external interface requirements. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 1	**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	The requirements matrix is incomplete and does not show the relationship of software requirements to the software architecture down to the unit level and to the tests used to verify the requirements. The requirements need to be uniquely identified as functional, performance, design constraints, attributes and external interface requirements. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 1	**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 is included as Appendix D within the SAPHIRE Version 8 Software Verification and Validation Plan Volume II.
N/A		
Criteria Priority: 2	**Does the Requirements Document identify the purpose and scope? NUREG/BR-0167 Section 2.2, 4.3**	
Pass		Comments
Fail	X	The requirements are included in section 2 Software Requirements and section 3 Interface Requirements Specification within the SAPHIRE Version 8 Software Verification and Validation Plan Volume I. There is no section addressing purpose or scope. There is no Software Requirements Documentation. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 2	**Does the Requirements Document identify what the products will and will not do? Software Engineering Practices**	
Pass		Comments
Fail	X	The requirements are included in section 2 Software Requirements and section 3 Interface Requirements Specification within the SAPHIRE Version 8 Software Verification and Validation Plan Volume I. There is no Software Requirements Documentation.
N/A		

		Refer to section 3.0 Summary of Findings.
Criteria Priority: 2	**Does the Requirements Document describe the objectives and goals? NUREG/BR-0167 Section 5.2.1**	
Pass		Comments
Fail	X	The requirements are included in section 2 Software Requirements and section 3 Interface Requirements Specification within the SAPHIRE Version 8 Software Verification and Validation Plan Volume I. There is no section addressing objectives and goals. There is no Software Requirements Documentation. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 2	**Does the Requirements Document describe any constraints on memory or other system constraints? Software Engineering Practices**	
Pass		Comments
Fail	X	The requirements are not uniquely identified. Multiple requirements are listed in paragraphs. Testability and traceability of the requirements through the software life cycle cannot be achieved. The requirements need to be uniquely identified as functional, performance, design constraints, attributes and external interface requirements. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 3	**Does the Requirements Document describe backup and recovery operations, if applicable? Software Engineering Practices**	
Pass		Comments
Fail	X	The requirements are included in section 2 Software Requirements and section 3 Interface Requirements Specification within the SAPHIRE Version 8 Software Verification and Validation Plan Volume I. There is no section addressing backup and recovery operations. There is no Software Requirements Documentation. Refer to section 3.0 Summary of Findings.
N/A		
Criteria Priority: 2	**Does the Requirements Document describe assumptions? (Assumptions can lead into Risks) Software Engineering Practices**	
Pass		Comments
Fail	X	The requirements are included in section 2 Software Requirements and section 3 Interface Requirements Specification within the SAPHIRE Version 8 Software Verification and Validation Plan Volume I. There is no section addressing assumptions. There is no Software Requirements Documentation. Refer to section 3.0 Summary of Findings.
N/A		