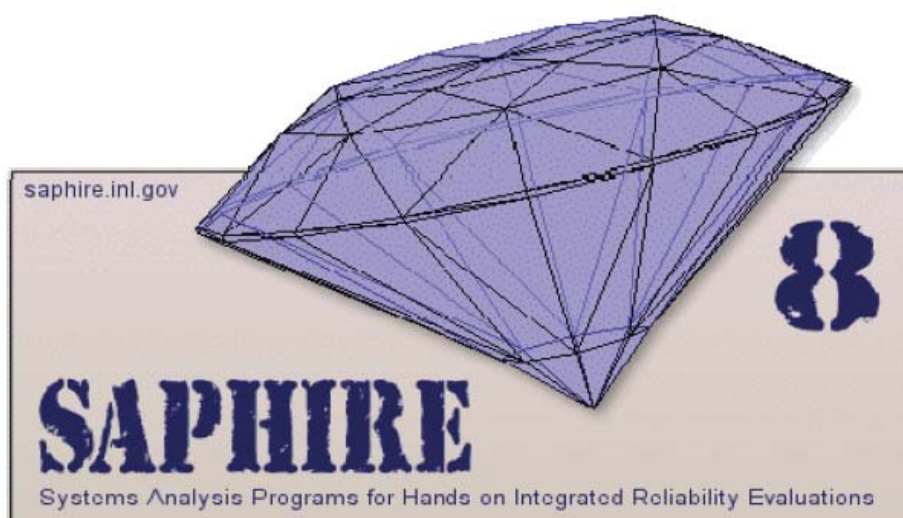


SAPHIRE 8 Software Acceptance Test Plan

July 2009



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

SAPHIRE 8 Software Acceptance Test Plan

July 2009

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

<http://www.inl.gov>

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

Idaho National Laboratory		Page: 1 of 77	
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier:	INL/EXT-09-16236
		Revision:	0
		Effective Date:	July 21, 2009

REVISION LOG

<u>Revision Number</u>	<u>Effective Date</u>	<u>Affected Pages</u>	<u>Description of Change</u>
0	07/21/2009	0	Initial issue

Idaho National Laboratory		Page: 2 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

CONTENTS

REVISION LOG.....	1
1. Introduction.....	1
1.1 Project Background and Objectives.....	1
1.2 Project Scope and Organization.....	2
2. Testing Approach.....	8
2.1 Features to be Tested	8
2.2 Features Not Tested	18
3. Test Methodologies.....	19
3.1 Background.....	19
3.2 Testing Approach.....	34
3.3 Test Data	49
3.4 Test Documents	49
3.5 Requirements Validation	49
3.6 Test Control Procedures.....	49
4. Test Phases.....	50
4.1 Test Design of the Automated Test Suite	51
4.2 Participants.....	Error! Bookmark not defined.
4.3 Sources of Data	60
4.4 Entrance and Exit Criteria.....	60
4.5 Requirements	60
4.6 Work Products	60

Idaho National Laboratory		Page: 3 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

5.	Development and Test Environment	61
5.1	Hardware Platform.....	61
5.2	Software Platform	61
5.3	Test Location	62
5.4	Staffing and Training	63
5.5	Test Summary	63
6.	Schedule.....	63
A-1.	Events and Condition Assessment Tests for Arkansas Nuclear 01(AN01).....	66
B-1.	Class Change Tests for DEMO Plant results	72
C-1.	Modula-2 Coding Standard.....	Error! Bookmark not defined.
C-2.	Delphi Coding Standard.....	Error! Bookmark not defined.

Idaho National Laboratory		Page: 1 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

1. Introduction

1.1 Project Background and Objectives

The Atomic Energy Act of 1975 replaced the U.S. Atomic Energy Commission with the U.S. Nuclear Regulatory Commission (NRC) and the U.S. Energy Research and Development Administration (ERDA, the predecessor to the Department of Energy). Since that time, the National Laboratories have had the mission to support the NRC with engineering services. Objective 7 of the INL Strategic Plan calls for the INL to “establish a vital energy security business.” The work for the NRC is a significant element of providing nuclear energy security for the nation.

This Software Project Plan (SPP) documents the technical work-scope that will be managed by the INL related to JCN N6423.

The NRC developed the SAPHIRE computer code for performing probabilistic risk assessments (PRAs) using a personal computer (PC) at the Idaho National Laboratory (INL) under Job Code Number (JCN) L1429. SAPHIRE started out as a feasibility study for a PRA code to be run on a desktop personal PC and evolved through several phases into a state-of-the-art PRA code. The developmental activity of SAPHIRE was the result of two concurrent important events: The tremendous expansion of PC software and hardware capability of the 90s and the onset of a risk-informed regulation era.

Three SAPHIRE versions have been released to date. Version 5 was a DOS version that became a production code a number of years ago. Version 6 was a Windows NT version that became a production code in 1998. Version 7 is also a Windows NT (or above) version that is currently the standard that is being used by the NRC.

Work began on a new version of SAPHIRE, Version 8, under JCN Y6394, “Maintenance and User Support for SAPHIRE Code and Library of PRAs.” Version 8 is being designed to meet current NRC program needs such as those related to SPAR model development, the Significance Determination Process (SDP) program, the Risk Assessment Standardization Project (RASP), as well as the Accident Sequence Precursor (ASP) Program. Development of Version 8 continued under JCN N6203, “Maintain and Support SAPHIRE Code and Library.” JCN N6423 “SAPHIRE Version 8” will support the beta and final Version 8, testing, verification, and validation.

The development of the new SAPHIRE version includes new features and capabilities. These features and capabilities are related to working with larger, more complex models and improving the user-friendliness of SAPHIRE’s interfaces while retaining key functionality of Version 7.

Version 8 is being developed to support the SPAR models and to run them as an integrated model (e.g., Level 1 with external events). The graphical user interface has also improved from SAPHIRE 7 to support NRC programs such as the SDP and the ASP. A tailored interface for the

Idaho National Laboratory		Page: 2 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

SDP and the ASP programs is being developed. An interface for general analyses and model construction is also being developed. The interfaces for the SDP, ASP, and general analysis introduce the concept of a “workspace” in which the analyst may run and save different analyses. The use of workspaces enables the user to separate the model construction from the model analysis.

Other features and capabilities added to the new version of SAPHIRE include the following. Size limitations have been removed on the number of basic events, fault trees, sequences, end states, etc. Work group integration capability has been added, allowing analysts to work on different model portions and integrate them to construct a large integrated model. Version 8 has also been tested on a multiple processor computer to study improving the speed of calculations.

In JCN N6423, a work plan will be developed to identify the work needed to complete a SAPHIRE 8 version for beta testing and to finalize the code for general release.

Projects JCN L1429 and JCN W6241 were closed out at the end of their periods of performance, November 30, 2000, for both projects. The remaining work from these two projects was consolidated into JCN Y6394, which has been closed out. JCN N6203 continues work on the current version of SAPHIRE, Version 7. Work is now being continued under JCN N6423 for SAPHIRE Version 8.

1.2 Project Scope and Organization

The INL may perform a wide range of engineering and research projects supporting all Offices under the NRC’s Executive Director for Operations. Areas of specific expertise that the NRC has historically relied upon the INL to provide include probabilistic risk assessment, reliability analysis, operational data collection and trending, operational event assessment, component engineering, and training. Support in new technical areas is constantly being pursued.

All NRC work assigned to a DOE national lab is governed by NRC Directive 11.7, NRC Procedures for Placement and Monitoring of Work with the Department of Energy. The NRC assigns each project a unique Job Code Number (JCN), is funded separately, and is assigned a NRC Project Manager and NRC Technical Monitor. NRC Directive 11.7 establishes a controlled and monitored process for requesting services of a national lab, work planning, work authorization and initiation, work progress monitoring, reporting, work termination and project closeout. The primary documents to accomplish these functions are: Request for Proposal with attached Statement of Work, NRC Form 189 (DOE Laboratory Project and Cost Proposal for NRC Work), NRC Form 173 (Standard Order for DOE Work), and the Monthly Letter Status Report.

A detailed description for each project is established, documented, and approved by the NRC and DOE-ID on NRC Form 189. A summary description of each project is provided on the Control Account. These two documents provide the official project descriptions. All other project

Idaho National Laboratory		Page: 3 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

management documents should refer to one of these two documents for a description.

The work scope for each JCN is provided in detail in the individual project's current NRC Form 189. All other project management documents should refer to the 189 for work scope.

The organizational structure of the SAPHIRE software development team influences and controls the software quality. Roles and responsibilities within the organizational structure provide the development team with the freedom, flexibility and objectivity to evaluate and monitor the software quality as well as verify problem resolutions. This structure enables the development team to tailor the maintenance and development activities, techniques, and methodologies for problem identification, reporting and resolution, testing, records retention, and configuration management.

For the INL, Software Quality Assurance (SQA) requirements are contract driven and interpreted from DOE Order 414.1C, "Quality Assurance", 10 CFR 830 Subpart A, "Nuclear Safety Management", and ASME NQA-1-2000, "Quality Assurance Requirements for Nuclear Facility Applications." The INL internal document, PDD-13610, "Software Quality Assurance Program," describes the SQA Program at the INL. To implement the SQA Program, two supporting documents are used at the INL:

1. LRD-13600, "Software Quality Assurance" specifies the requirements and responsibilities for controlling the quality of software and applies to laboratory organizations that develop, procure, modify, maintain, operate, use, or retire software.
2. LWP-13620, "Software Quality Assurance" is the entry-level document for the SQA work processes. This procedure directs the SQA activities during the software life cycle which consists of requirements, design, implementation, acceptance test, operations, maintenance, and retirement. LWP-13620 is designed to standardize the lab's SQA implementation by applying a graded approach and utilizing trained personnel in the identification of the required SQA activities. LWP-13620 provides direction in determining the rigor (level of effort) required when (a) performing SQA activities and (b) creating documentation for each phase of the software life cycle.

SQA must be applied to all INL software (including SAPHIRE development) activities that meet the criteria in LWP-13620. Performing SQA is important because it (a) maintains compliance with DOE O 414.1C, "Quality Assurance" and 10 CFR 830 "Nuclear Safety Management", Subpart A "Quality Assurance Requirements"; (b) assists in assuring you are following a stable, repeatable process that is cost effective and consistently meets customer requirements; and (c) provides a foundation for ensuring the quality of software developed, procured, and modified at the INL.

Per LWP-13610, the Software Owner is a representative of the organization responsible for the application of the software. He/she is identified in the INL Enterprise Architecture and is responsible for:

Idaho National Laboratory		Page: 4 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

- Identifying and documenting the appropriate safety software categorization to software per LWP-13620.
- Approving software management plan, requirement, acceptance test, and retirement documentation
- Approving results of evaluations of acquired and legacy software to determine adequacy to support the operations, maintenance, and retirement phases
- Procuring software using LWP-4001, "Material Acquisitions" and LWP-4002, "Service Acquisitions."
- Developing and implementing program-specific training for the operational use of safety software
- Considering whether user training is needed for the operational use of Quality Level 1 and Quality Level 2 software.

"Software" as defined by the PDD-13610 procedure pertains to computer programs and associated documentation and data pertaining to the operation of a computer system and includes:

1. Application Software - software designed to fulfill specific needs of a user; for example navigation, payroll, or process control.
2. Support Software such as the following software tools (e.g., compilers, configuration and code management software, editors) or system software (e.g., operating systems).

Note that within the INL SQA process, software that does not fall within the scope of the SQA Program includes any software covered by a contractual agreement, such as Work for Others, that includes references or requires a specific documented SQA process. Currently, the SAPHIRE development does not reference or require a specific documented SQA process – thus the development falls under the INL SQA process.

It is the responsibility of the **Software Owner** to make the determination as to whether a particular software can be classified as "safety software." Safety Software includes the following type of software:

- **Safety System Software.** Software for a nuclear facility that performs a safety function as part of a structure, system, or component and is cited in either (a) a DOE approved documented safety analysis or (b) an approved hazard analysis per DOE P 450.4, Safety Management System Policy, dated 10-15-96, and the DEAR clause.
- **Safety Analysis and Design Software.** Software that is used to classify, design, or analyze nuclear facilities. This software is not part of a structure, system, or

Idaho National Laboratory		Page: 5 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

component (SSC) but helps to ensure that the proper accident or hazards analysis of nuclear facilities or an SSC that performs a safety function.

- **Safety Management and Administrative Controls Software.** Software that performs a hazard control function in support of nuclear facility or radiological safety management programs or technical safety requirements or other software that performs a control function necessary to provide adequate protection from nuclear facility or radiological hazards. This software supports eliminating, limiting or mitigating nuclear hazards to worker, the public, or the environment as addressed in 10 CFR 830, 10 CFR 835, and the DEAR ISMS clause.

For all software that falls within the scope of the SQA Program, a **quality level** must be assigned by a qualified Quality Level Analyst with review and concurrence by a Quality Level Reviewer (i.e., a second Quality Level Analyst) per LWP-13014, "Determining Quality Levels." The Quality Level Analyst should then communicate to the Software Owner the determined quality level.

However, for software deployed prior to the release of the revised INL SQA Program (SAPHIRE development falls into this category):

1. When the revised INL SQA Program becomes effective (3/29/2007), the date for completion of the QL determination for legacy software projects must be identified and documented.
2. At the time the software is modified, the QL determination activity must be performed.
3. ALL software must have an associated QL Determination by no later the 3/31/2008.

A quality level is a designator that identifies the relative risk associated with the failure of items or activities. This quality level determination must be performed regardless of the size or complexity of the software. The Quality Levels are defined as follows:

Quality Level 1 software is software whose failure creates "high" risk. This software requires a high degree of rigor during the software life cycle.

Quality Level 2 software is software whose failure creates "medium" risk. This software requires a moderate degree of rigor during the software life cycle.

Quality Level 3 software is software whose failure creates "low" risk. This software requires a low degree of rigor during the software life cycle.

The quality level of the software is a component when determining the level of rigor (graded-approach) that the SQA Specialist must ensure is applied when performing software quality assurance activities or creating documentation for each phase of the software life cycle. The

Idaho National Laboratory		Page: 6 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

higher the quality level of the software, the more rigorous the quality assurance activities and documentation will need to be as defined in Section 4.2 of LWP-13620.

Per the Requirements Phase Documentation table in LWP-13620:

1. For QL-1 and QL-2 Custom-Developed or Configurable software: include within the software's documentation (e.g., Project Execution Plan (PEP), Software Quality Assurance Plan (SQAP)):
 - a. The activities to be performed to support software quality assurance (e.g., design reviews, acceptance testing, reviews and audits),
 - b. Identify SQA documentation to be generated,
 - c. Identify the roles and responsibilities for SQA activities, and
 - d. The methodology for tracing requirements throughout the software life cycle.
2. For QL-3 Custom Developed or Configurable software and all other software types (i.e., acquired, utility calculations, commercial D&A), the described information is optional or not applicable within the software's documentation.

All documentation that furnishes evidence of the software quality is considered a QA record and should be handled as a quality record according to the organization, program, or project's Records Management Plan as required by LWP-1202. QA records generated during the software development life cycle could include project plans, requirement specifications, configuration management plans, software quality assurance plans, security plans, and verification and validation documentation (e.g., test plans, test cases, design review documents).

For the SAPHIRE 8 development project, the INL-derived QA program (LWP-13620) quality level has been set at Quality Level 3.

Note that the quality of NRC research programs are assessed each year by the Advisory Committee on Reactor Safeguards. Within the context of their reviews of RES programs, the definition of quality research is based upon several major characteristics:

- 75% Results meet the objectives
- 17% Justification of major assumptions (12% of total)
- 68% Soundness of technical approach and results (52% of total)
- 15% Uncertainties and sensitivities addressed (11% of total)
- 25% Documentation of research results and methods is adequate

Idaho National Laboratory		Page: 7 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

64% Clarity of presentation (16% of total)

36% Identification of major assumptions (9% of total)

It is the responsibility of the contractor to ensure that these quality criteria are adequately addressed throughout the course of the research that is performed. The NRC PM and TM will review all research products with these criteria in mind.

INL will follow NRC Management Directive 11.7 “Procedures for Placement and Monitoring of Work with the Department of Energy” related to software development. This directive suggests that “all software development, modification, or maintenance tasks shall follow general guidance provided in NUREG/BR-0167 “Software Quality Assurance Program and Guidance.” SAPHIRE 8 will follow the requirements for Level 1 software defined in Section 1.2 of NUREG/BR-0167.

The NRC will perform an audit of the software QA implementation one a year against the requirement of NUREG/BR-0167.

Idaho National Laboratory		Page: 8 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

2. Testing Approach

2.1 Features to be Tested

To determine the SAPHIRE features most important to be tested, the critical tasks performed in a PRA (e.g., fault tree analysis, event tree analysis, sensitivity analysis) were identified. Then the SAPHIRE functions needed to accomplish each of these tasks were determined. The review process produces a list of items to be tested, which PRA analysis experts using SAPHIRE reviewed and revised. In summary, the following SAPHIRE functions are tested:

- Fault Tree Analysis, including cut set generation and quantification, application of recovery rules (i.e., modifications made to the cut set results after they are generated), and the capability to perform the analysis on a single fault tree or on multiple fault trees
- Event Tree and Sequence Analysis, including event tree sequence generation, sequence cut set generation, quantification, application of recovery rules, application of partition rules (i.e., steps to move particular cut sets to a specified end state category), and the capability of performing the analysis on a single event tree/sequence or on multiple trees/sequences
- End State Analysis, including gathering cut sets by sequence end-state designation, gathering of cut sets by partitioning rules, quantification, and the capability of performing the analysis on a single end state or on multiple end states
- Importance Measures Analysis, for options available to quantify importance measures
- Uncertainty Analysis, for individual sequences or groups of sequences, using either Latin Hypercube or the Monte Carlo sampling
- Change Sets, and other similar features, providing the capability to perform sensitivity analyses (change sets contain user-defined modifications to basic event probabilities)
- Data Utility Functions intended to facilitate data handling and manipulation.
- Mapping of systems, components, failure modes to basic events and then using them in sequence analysis
- Linking event trees with the “Generate cut set” option checked used for the “large event tree” PRA methodology
- Transformations, which are generally used only for fire or flooding analyses
- Seismic analysis
- Stress testing (record count up to 2 million sequences).
- Verification of the new user interface. (Manual tests at this point in time).

**Software Acceptance Test Plan
for SAPHIRE Version 8 N6423**

Identifier: INL/EXT-09-16236
Revision: 0
Effective Date: July 21, 2009

To test the above SAPHIRE functions, a variety of models were selected, with varying degrees of size and complexity, based on their suitability for adequately testing the selected functions. The intent of this effort was to acquire basic assurance that new updates or changes have not compromised any existing capabilities. Size and boundary conditions of the PRA models were not major issues. Databases of typical size and complexity were selected from among the available models. A decision was made to test more features with less complex models than to test fewer features with complex models. Actual PRA models developed by experienced analysts for analyzing nuclear power plants were used for the tests. Two tests: importance measures and change sets were conducted on artificial plant models. Table 1 shows the functions tested by each type of SAPHIRE project. At a minimum, each feature tested was evaluated with at least two PRA models. Further, many of the basic features (e.g., moving from one menu option to the next, basic event probability generation, and minimal cut set solving) were tested by almost all the PRA models owing to the need to perform these basic functions as part of a more complex calculation.

Idaho National Laboratory		Page: 10 of 77	
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier:	INL/EXT-09-16236
		Revision:	0
		Effective Date:	July 21, 2009

Table 1. Tests where specific PRA features are verified.

Test	Test Name	PRA Area	SAPHIRE Option	Test Models
Test-01	Solve Fault Trees Fault Tree Probability Results.	Generate current event data	No change set data	All
Test-02	Solve Sequences Core Damage Frequency Results.			
Test-05	Transient initiator with no other failures.			
Test 06	Small LOCA initiator with no other failures.			
Test-07	Steam Generator initiator with no other failures.			
Test-08	Grid-related LOOP initiator with no other failures.			
Test-09	Plant-centered LOOP initiator with no other failures.			
Test-10	Severe Weather LOOP initiator with no other failures.			
Test-11	Extreme Severe Weather LOOP initiator with no other failures.			
Test-13	Project Uncertainty.			
Test-14	Log Normal Distribution using MCS.			
Test-15	Normal Distribution MCS.			
Test-16	Beta Distribution MCS.			
Test-17	Chi-squared Distribution MCS.			
Test-18	Exponential Distribution MCS.			
Test-19	Uniform Distribution MCS.			
Test-20	Gamma Distribution MCS.			
Test-21	Maximum Entropy Distribution MCS.			
Test-23	Seismic Log Normal Distribution MCS.			
Test-24	Constrained Non-informative Distribution MCS.			
Test-25	Log Normal Distribution using LHS.			
Test-26	Normal Distribution LHS.			
Test-27	Beta Distribution LHS.			
Test-28	Chi-squared Distribution LHS.			
Test-29	Exponential Distribution LHS.			
Test-30	Uniform Distribution LHS.			
Test-31	Gamma Distribution LHS.			
Test-32	Maximum Entropy Distribution LHS.			
Test-34	Seismic Log Normal Distribution LHS.			
Test-35	Constrained Non-informative Distribution LHS.			
Test-36	Histogram Distribution MCS.			
Test-37	Histogram Distribution LHS.			
Test-38	Gather End States.			

Idaho National Laboratory		Page: 11 of 77	
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236	
		Revision: 0	
		Effective Date: July 21, 2009	

Test	Test Name	PRA Area	SAPHIRE Option	Test Models
Test-39	End State Single/Group Uncertainty MCS.			
Test-40	End State Single/Group Uncertainty LHS.			
Test-42	Link Level 1 Event Trees (small event trees).			
Test-43	Partition Sequence Cut Sets.			
Test-44	Link PDS Event Trees (large event trees)..			
Test-45	Fault Tree Importance Measures.			
Test-46	Sequence Importance Measures.			
Test-47	Sequence Group Importance Measures.			
Test-48	End State Importance Measures.			
Test-49	End State Group Importance Measures.			
Test-03	Events and Condition Assessment - MFW unavailable for 72 hours.	Generate current event data	Single changes	DEMO, SURRY-50
Test-04	Emergency Diesel Generator out of Service for 3 months.			
Test-12	Transient initiator with AFW failed.			
Test-22	Dirichlet Distribution MCS.			
Test-33	Dirichlet Distribution LHS.			
Test-50	Single Change Set on Compound Event.			
Test-50	Single Change Set on Compound Event	Generate current event data	Single changes	Wolf Creek, Peach Bottom
Test-51	Change Set Processing- Class. Class change - all events, probability 0.1 Class change - ?-MOV-1 events, probability 0.5 (a subset)	Generate current event data	Class changes	DEMO, SURRY-50
Test-52	Change Set Processing-Marked Order Marked change sets from scenarios 1, 2, 3 (marked in that order)	Generate current event data	Marked order	DEMO, SURRY-50
Test-41	Cut Set Verification Cut Set Verification Solve Sequences and Monte Carlo uncertainty calculations. Cut Set Verification Solve Fault Trees, Sequences and end states.	Fault tree cut set generation	With flag sets	COM-PEAK, SURRY-50, SEQUOIA
Test-01	Solve Fault Trees Fault Tree Probability Results	Fault Tree cut set generation	Without flag sets	SPAR, ** COM-PEAK,
Test-41	Cut Set Verification			SURRY-50, CR3
Test-53	Basic Events Load / Extract Fault Tree Load / Extract			
Test-41	Cut Set Verification	Sequence cut set Generation	With flag sets	All
Test-02	Solve Sequences Core Damage Frequency Results	Sequence cut set Generation	Without flag sets	Multiple

Idaho National Laboratory		Page: 12 of 77	
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier:	INL/EXT-09-16236
		Revision:	0
		Effective Date:	July 21, 2009

Test	Test Name	PRA Area	SAPHIRE Option	Test Models
Test-13 Test-41 Test-42	Project Uncertainty Cut Set Verification			
Test-38 Test-41 Test-44	Gather End States Cut Set Verification Link PDS Event Trees (large event trees).	Gather sequence cut sets into end states	By sequence	BV2-5, SURRY-50, COM-PEAK, S_LERF
Test-43	Partition Sequence Cut Sets.	Gather sequence cut sets into end states	By Cut Set	S_LERF
Test-14 Test-21 Test-23 Test-24	Log Normal Distribution using MCS Maximum Entropy Distribution Seismic Log Normal Distribution Constrained Non-informative Distribution	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-14 Test-21 Test-23	Log Normal Distribution using MCS Maximum Entropy Distribution Seismic Log Normal Distribution	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-14 Test-21 Test-23	Log Normal Distribution using MCS Maximum Entropy Distribution Seismic Log Normal Distribution	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-14 Test-21 Test-23	Log Normal Distribution using MCS Maximum Entropy Distribution Seismic Log Normal Distribution	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-14 Test-21 Test-23	Log Normal Distribution using MCS Maximum Entropy Distribution Seismic Log Normal Distribution	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-14 Test-21 Test-23	Log Normal Distribution using MCS Maximum Entropy Distribution Seismic Log Normal Distribution	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-14 Test-21 Test-23	Log Normal Distribution using MCS Maximum Entropy Distribution Seismic Log Normal Distribution	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-14 Test-21 Test-23	Log Normal Distribution using MCS Maximum Entropy Distribution Seismic Log Normal Distribution	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-14 Test-21 Test-23	Log Normal Distribution using MCS Maximum Entropy Distribution Seismic Log Normal Distribution	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-14 Test-21 Test-23	Log Normal Distribution using MCS Maximum Entropy Distribution Seismic Log Normal Distribution	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-14 Test-21 Test-23	Log Normal Distribution using MCS Maximum Entropy Distribution Seismic Log Normal Distribution	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-14 Test-21 Test-23	Log Normal Distribution using MCS Maximum Entropy Distribution Seismic Log Normal Distribution	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU

Idaho National Laboratory		Page: 13 of 77	
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236	
		Revision: 0	
		Effective Date: July 21, 2009	

Test	Test Name	PRA Area	SAPHIRE Option	Test Models
Test-23	Seismic Log Normal Distribution			
Test-14	Log Normal Distribution using MCS	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-21	Maximum Entropy Distribution			
Test-23	Seismic Log Normal Distribution			
Test-14	Log Normal Distribution using MCS	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-21	Maximum Entropy Distribution			
Test-23	Seismic Log Normal Distribution			
Test-14	Log Normal Distribution using MCS	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-21	Maximum Entropy Distribution			
Test-23	Seismic Log Normal Distribution			
Test-14	Log Normal Distribution using MCS	Uncertainty of fault tree distributions	Monte Carlo sampling	TSTU
Test-21	Maximum Entropy Distribution			
Test-23	Seismic Log Normal Distribution			
Test-36	Histogram Distribution-MCS			
Test-25	Log Normal Distribution using LHS	Uncertainty of fault tree distributions	Latin Hypercube sampling	TSTU
Test-26	Normal Distribution using LHS			
Test-27	Beta Distribution LHS			
Test-28	Chi-squared Distribution LHS			
Test-29	Exponential Distribution LHS			
Test-30	Uniform Distribution LHS			
Test-31	Gamma Distribution LHS			
Test-32	Maximum Entropy Distribution LHS			
Test-33	Dirichlet Distribution LHS			
Test-34	Seismic Log Normal Distribution LHS			
Test-35	Constrained Non-informative Distribution LHS			
Test-37	Histogram Distribution-LHS			
Test-22	Dirichlet Distribution MCS	Sequence uncertainty analysis	Monte Carlo sampling	TSTU
Test-24	Constrained Non-informative Distribution MCS			
Test-38	Gather End States	Gathering of End States		BV2-5
Test-39	End State Single/Group Uncertainty MCS	End State uncertainty analysis	Monte Carlo sampling	BV2-5
Test-40	End State Single/Group Uncertainty LHS	End State uncertainty analysis	Latin Hypercube sampling	BV2-5
Test-45	Fault Tree Importance Measures Fault Tree Fussell-Vesely Importance (ratio)	Importance measures	Fault trees	DEMO

Idaho National Laboratory		Page: 14 of 77	
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier:	INL/EXT-09-16236
		Revision:	0
		Effective Date:	July 21, 2009

Test	Test Name	PRA Area	SAPHIRE Option	Test Models
	Fault Tree Birnbaum Importance (Interval or difference) Fault Tree Uncertainty Importance			
Test-46	Sequence Importance Measures Sequence Fussell-Vesely Importance (ratio). Sequence Birnbaum Importance (interval or difference). Sequence Uncertainty Importance.	Importance measures	Sequence	DEMO
Test-47	Sequence Group Importance Measures Sequence Group Fussell-Vesely Importance (ratio). Sequence Group Birnbaum Importance (interval or difference). Sequence Group Uncertainty Importance.	Importance measures	Sequence Group	BV2-5
Test-48	End State Importance Measures End State Fussell-Vesely Importance End State Birnbaum Importance End State Uncertainty Importance	Importance measures	End State	HISNO, BV2-5
Test-49	End State Group Importance Measures End State Group Fussell-Vesely Importance End State Group Birnbaum Importance End State Group Uncertainty Importance	Importance measures	End State Group	BV2-5
Test-41	Cut Set Verification	Cut Set Update	Fault trees	SURRY-50, COM-PEAK, CR3
Test-53	Basic Events Load / Extract Fault Tree Load / Extract	Uncertainty analysis		
Test-13	Project Uncertainty			
Test-41	Cut Set Verification	Cut Set Update	Fault trees	SURRY-50, COM-PEAK, CR3
Test-41	Cut Set Verification	Fault tree cut set Recovery	Auto-recover option	SURRY-50, COM- PEAK, CR3
Test-02	Solve Sequences Core Damage Frequency Results	Sequence cut set Recovery	Auto-recover option	SURRY-50, COM- PEAK
Test-13	Project Uncertainty			
Test-41	Cut Set Verification	Sequence cut set Partitioning	Batch apply option	S_LERF
Test-43	Partition Sequence Cut Sets.	Link Small event tree (logic)	Linkage Rules	S_LERF
Test-42	Link Level 1 Event Trees (small event trees).	Link Large event tree (cut)	Create cut sets option	S_LERF
Test-44	Link PDS Event Trees (large event trees).			

Idaho National Laboratory		Page: 15 of 77	
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier:	INL/EXT-09-16236
		Revision:	0
		Effective Date:	July 21, 2009

Test	Test Name	PRA Area	SAPHIRE Option	Test Models
		sets)		
Test-41	Cut Set Verification	Fault Tree logic	Alpha-numeric logic editor	SURRY-50, COM-PEAK
Test-41	Cut Set Verification	Fault Tree logic	Graphical editor	SURRY-50, COM- PEAK, CR3
Test-54	Fault Tree Utilities: Auto Page/Solve Fault Tree Utilities: Cut Sets to End State	Fault Tree Logic	Pager	CR3
Test-42	Link Level 1 Event Trees (small event trees)	Event tree logic	Graphical editor	S LERF
All tests		Project version control	Version Upgrade	All Models
Test-05	Transient initiator with no other failures.	Initiating Event Assessments	Delete	All Models
Test-06	Small LOCA initiator with no other failures.			
Test-07	Steam Generator initiator with no other failures.			
Test-08	Grid-related LOOP initiator with no other failures.			
Test-09	Plant-centered LOOP initiator with no other failures.			
Test-10	Severe Weather LOOP initiator with no other failures.			
Test-11	Extreme Severe Weather LOOP initiator with no other failures.			
Test-12	Transient initiator with AFW failed.			
Test-05	Transient initiator with no other failures.	Initiating Event Assessments	Add	All Models
Test-06	Small LOCA initiator with no other failures.			
Test-07	Steam Generator initiator with no other failures.			
Test-08	Grid-related LOOP initiator with no other failures.			
Test-09	Plant-centered LOOP initiator with no other failures.			
Test-10	Severe Weather LOOP initiator with no other failures.			
Test-11	Extreme Severe Weather LOOP initiator with no other failures.			
Test-12	Transient initiator with AFW failed.			
Test 03	Events and Condition Assessment - MFW unavailable for 72 hours	Events and Condition Assessments	Delete	All Models
Test-04	Emergency Diesel Generator out of Service for 3 months			
Test 03	Events and Condition Assessment - MFW unavailable for 72 hours	Events and Condition Assessments	Add	All Models
Test-04	Emergency Diesel Generator out of Service for 3 months			
Test 03	Events and Condition Assessment - MFW unavailable for 72 hours	Events and Condition Assessments	Add events to Assessment	All Models
Test-04	Emergency Diesel Generator out of Service for 3 months			

Idaho National Laboratory		Page: 16 of 77	
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier:	INL/EXT-09-16236
		Revision:	0
		Effective Date:	July 21, 2009

Test	Test Name	PRA Area	SAPHIRE Option	Test Models
Test-03	Events and Condition Assessment - MFW unavailable for 72 hours	Events and Condition Assessments	Process	All Models
Test-04	Emergency Diesel Generator out of Service for 3 months			
Test-53	Basic Events Load / Extract Fault Tree Load / Extract	Fault Trees	Load/Extract	CR3
Test-53	Basic Events Load / Extract Fault Tree Load / Extract	Basic Events	Load/Extract	CR3
Test-54	Fault Tree Utilities: Auto Page/Solve Fault Tree Utilities: Cut Sets to End State	Fault Trees	Modify/Delete	CR3
Test-55	Link Level 1 Event Trees (small event trees). Link Level 1 Event Trees: Solve w/ Flag Sets Solve Sequence Cut Sets w/ no Flags.	Level 1 Event Tree Linking	Linkage rules	Wolf Creek 302, Peach Bottom 302, SIMPLE-FT
Test-56	End-State Gathering	End States	End State Gathering	S_LERF (by rules), Beaver Valley (by names)
Test-57	Compound Event Plug ins	Common cause module, Utility module (i.e., add, multiply), Load-capacity	Fault Tree, Compound Event Plug-in	SiMPLE-FT (PLUG-IN-FT)
Test-58	Base Case Updates {*This test suite was never run *}	Base Case Update	Base Case Update	All SPAR 2Q, 3i models
Test-59	Calculation Types & N of M Gates Calculation Type True, N of M Gates Calculation Type False, N of M Gates Calculation Type Ignore, N of M Gates	Calculation types	True, 1,3,5,7, False, 1,3,5,7 Ignore, 1,3,5,7 Use of AND gates, then OR gates.	SiMPLE-FT
Test-60	Change Sets		Change Sets	TBD
Test-61	Uncertainty analysis		Uncertainty Distributions	TSTU, SUR40, Wolf Creek, Peach Bottom, SiMPLE-FT
Test-59	Calculation Types & N of M Gates Calculation Type True, N of M gates Calculation Type False, N of M gates Calculation Type Ignore, N of M gates	N of M Gates	Use of all inputs	SiMPLE-FT
Test-59	Load Capacity Test Calculation Type True, N of M Gates	Sequence generation	Sequence Generation	SiMPLE-FT (BE-LOAD-

Idaho National Laboratory		Page: 17 of 77	
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236	
	Revision:	0	
	Effective Date:	July 21, 2009	

Test	Test Name	PRA Area	SAPHIRE Option	Test Models
				CAPACITY)
Test-64	DOES NOT EXIST YET	Common Cause Failures, Basic Events with change sets	Common Cause Plug-ins	Wolf Creek, Peach Bottom 3
Test-65	Event Transformations	Basic Events, Sequence Analysis	Transformations	TBD
Test-66	Wrong Results	Results verification	Results verification	DEMO

Idaho National Laboratory		Page: 18 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

2.2 Features Not Tested

Like most software-development projects, time and budget constraints prohibited exhaustive testing. The verification effort focused on quantitative aspects of SAPHIRE. While the tests and acceptance criteria address a large part of the calculative functionality within SAPHIRE, the tests do not cover 100% of SAPHIRE capabilities. For example, the current test suite did not encompass every possible way of modifying cut sets after generation. Users can manipulate cut sets after generation (e.g., "post-processing") by manually editing them, using "recovery rules," using the "prune" option, and performing a cut set update. But the test suite does test the most commonly used mechanisms of performing tasks in SAPHIRE. Other calculative aspects not tested include the following:

- Event probability cut off (not frequently used due to the calculation speed of modern computers and software such as SAPHIRE)
- Solving sequences without fault trees (an obsolete calculation technique that may be removed from future versions of the software)
- Starting gate name (generally used only during development)

Idaho National Laboratory		Page: 19 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

3. Test Methodologies

3.1 Background

Early versions of SAPHIRE comprised a suite of programs that included the Models and Results Database (MAR-D); the Integrated Reliability and Risk Analysis System (IRRAS); the Systems Analysis Risk Assessment (SARA); and the Fault Tree, Event Tree, and Piping and Instrumentation Diagram (FEP) graphical editor. With Version 6.0, these programs were combined into a single program referred to simply as SAPHIRE. A new capability was added to SAPHIRE 6, designed for evaluating operational occurrences using the ASP interface. This new module was referred to as GEM.

Because use of SAPHIRE in regulatory applications is extensive, SAPHIRE is being tested through various processes. Earlier versions (SAPHIRE 4.0 and 5.0) were tested through a traditional (i.e., manual) verification and validation (V&V) process that met the IEEE's "Standard for Software Verification and Validation Plans (IEEE 1012-1986). Note that the user documentation and the software requirements specification were not evaluated to the detail recommended in the IEEE standard. Instead, the focus of the earlier V&V was solely on functionality of the software

Historically, each new version was beta tested to some degree before its release. Beta testers are analysts experienced with PRA methods and terminology and typically are familiar with earlier versions of SAPHIRE. The primary objective of the beta testing is to verify that the results produced by the new version are correct. The secondary objective is to ensure the software is user-friendly and functional. In addition, INL personnel receive feedback from users around the world. Hundreds of users rely on the calculative ability of the tool for both risk and reliability calculations. Included among these users are U.S. national laboratory personnel, U.S. and foreign government regulators, private contractors, university professors and students, industry researchers, and nuclear power plant PRA analysts. Based on feedback from users, programmatic errors and discrepancies are corrected, the user interface is improved, and new features are recommended.

Also, since the INL is continually loading new PRA models into SAPHIRE, new releases are tested extensively by (a) comparing them with PRA models and results of earlier versions and, especially, with PRAs loaded in version 5.0, and (b) by loading new PRAs and comparing them with the original PRA models and results. About 30 nuclear power plant PRAs have been (completely or partially) loaded into the toolset. The loading process requires a detailed comparison of the models and results with the models and results of the original PRAs. Identified discrepancies are resolved through interaction between individual licensees and PRA analysts at the INL. Given that different PRAs have been performed with different types of software, one can argue that SAPHIRE has been tested with an enormous number of test cases.

3.1.1 Overview of PRA Concepts

3.1.1.1 PRA Analysis Levels

A full-scope PRA involves three levels (see Figure 6-1). A modern nuclear power plant PRA is analyzed using a layered approach. The first level contains the logic models (e.g., fault trees and event trees) and probability data representing the outcome of damage to the reactor core. The second level concerns the plant response to the core damage progression (primarily the containment and associated

Software Acceptance Test Plan for SAPHIRE Version 8 N6423

Identifier: INL/EXT-09-16236

Revision: 0

Effective Date: July 21, 2009

systems). And third level focuses on the off-site consequences resulting from the damaged core and containment. These levels are called Level 1, Level 2, and Level 3 analysis, respectively (NRC 1989).

Level 1 PRA identifies and quantifies sequences leading to core damage. This involves identifying significant initiating events, generally those that challenge normal plant operation and that must be successfully dealt with to prevent core damage. Once these initiators are identified, possible plant responses (sequences) must be determined. The response depends on the different combinations of successes and failures of the systems involved. When the systems have been determined, they must be modeled (usually with fault trees) to identify credible failure modes and unavailability. Finally, a Level 1 PRA quantifies the plant's core damage frequency and its associated statistical uncertainty. To determine these results, initiating event frequencies and equipment failure/unavailability probabilities must be ascertained.

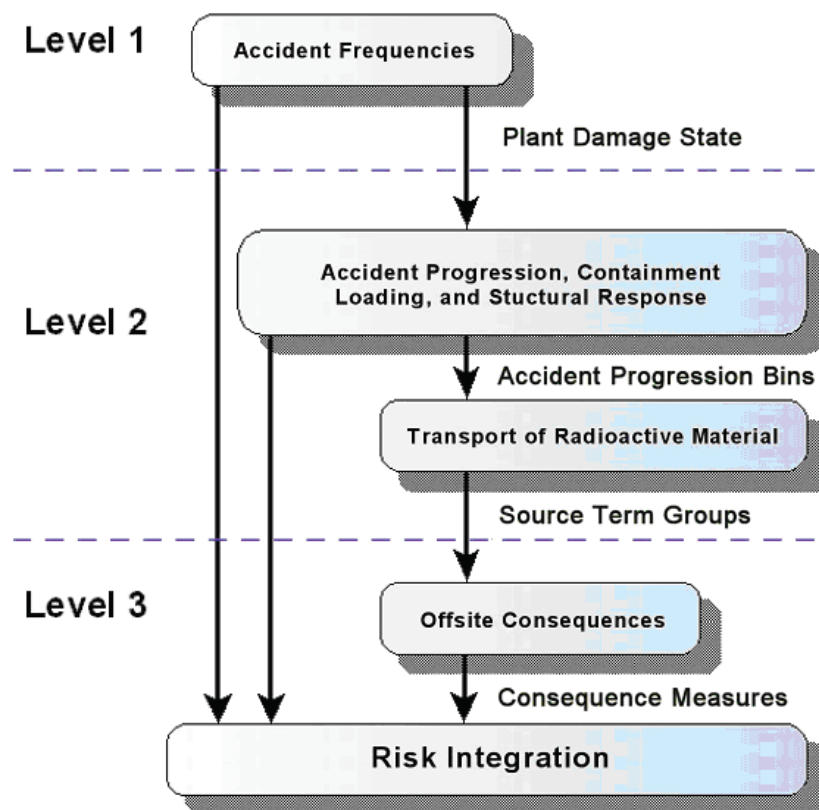


Figure 6.1 The Three Levels of a Full-Scope PRA.

A Level 2 PRA evaluates and quantifies subsequent material releases from core damage. This analysis involves filtering the large number of Level 1 sequences to a practical number for detailed analysis, typically by grouping Level 1 cut sets into a smaller set of plant damage states. Assessment of

Idaho National Laboratory		Page: 21 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

containment system performance with Level 1 accident sequence analyses is performed much the same as Level 1 analysis by using fault tree models to estimate failure probabilities and linking them directly to sequences developed in the Level 1 analysis.

Level 3 addresses not only Level 1 and 2 issues but evaluates and quantifies resulting consequences to the public and environment as well. Thus, questions such as weather conditions, population levels surrounding the plant site, and dispersion (from containment) characteristics are important in this analysis.

3.1.1.2 PRA Elements Embodied within SAPHIRE 8.0

The toolset provides mechanisms to model all traditional Level 1 tasks. Event trees can be built to determine accident sequences using initiating events and systems. The individual systems as named on the event trees can be modeled using the SAPHIRE graphic or logic fault tree editors. Initiating events and other failure events that comprise each system can be assigned frequencies or probabilities. Minimal cut sets (i.e., a minimally sufficient group of failures that can lead to an undesired outcome) can be generated to quantify fault trees and sequences. The PRA analyst has mechanisms available to perform a variety of different uncertainty analyses, sensitivity analyses, and importance measures.

In addition to its Level 1 capabilities, SAPHIRE has the functionality to perform many calculations for Level 2 analysis. Excluded from the built-in SAPHIRE functionality are complex phenomenological calculations (e.g., post core damage situations) germane to Level 2 analyses. SAPHIRE also offers the analyst some simple Level 3 calculation tools, but these focus on logic-model scenario development (e.g., containment is breached with probability X and the dispersion has a magnitude of Y). In general, the toolset is intended to provide analysts a tool for developing and quantifying Level 1 and 2 PRA models.

Since the software is a tool primarily to conduct Level 1 and 2 PRA analyses, our testing of this software focused on the important areas relating to this type of PRA. Consequently, the tests themselves and the functional areas of the software being subjected to examination are defined in terms of PRA attributes. The remainder of this section discusses these PRA attributes and specific SAPHIRE terms.

Within the three PRA levels of analysis are a variety of computational models, data, and associated assumptions. The PRA elements described herein are an Initiating Event, an Accident sequence, Event trees, End states, and Success criteria.

An initiating event is an operational occurrence (such as a pipe break or plant transient) which threatens fuel safety and may require safety system response to avoid core damage.

For an Accident sequence, the branching paths, from initiating event to end state, that is a unique combination of system failures and/or successes. This string of failures or successes defines the sequence of events leading to an accident (i.e., undesired end state). The sequence results are expressed as minimal cut sets.

Event trees are logic models that are used to identify accident sequences. Within an event tree, all sequences begin from an identical initiating event. After the initiating event, specific system responses determine the course of the accident sequence.

Idaho National Laboratory		Page: 22 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

End states are groups of accident sequences, which share certain characteristics that the analyst delineates. These groups may be related to ability to perform selected safety functions or timing of functional failures.

The success criteria are a specification of the compliment of equipment that must successfully operate to achieve functional success for each branch point or top event. This specification is developed analytically. The success criteria are used to define the boundary conditions for the failure modeling embodied within the corresponding fault tree.

3.2 Plant Models Available In SAPHIRE 8.0

Plant models become available from two major sources. Models of various levels of detail arise from the Plant Database Development for the SAPHIRE project, which converts existing PRA models from other software to also operate in SAPHIRE. These models are listed in Table 2.

Table 2. Basic plant models.

Item	Plant, Design, Region	IRRAS/SAPHIRE Version, Database Types(s), ASP Category	PRA Source, Date, Type	Comments
1	BRUNSWICK UNIT 1 General Electric Type 4 Mark 1 Containment Region II	SAPHIRE 6.0 model only. BNP1-FTL Fault Tree Level 1 load At power conditions. ASP Category: BWR C	Carolina Power & Light Company's IPE (1992), plus updated data Modeling Methodology: Master Logic Diagram (Large Fault Tree Model) SAIC's CAFTA Code	Database completed September 17, 1998.
2	COMANCHE PEAK UNITS 1 & 2 Westinghouse Electric 4-Loop Dry, Ambient Pressure Containment Region I	SAPHIRE 6.0 model only. COM- PEAK Full scope Level 1 load: Linked Event Tree and Fault Tree Model At power conditions. ASP Category: PWR B	Texas Utilities Electric Company IPE (1992) Modeling Methodology: Small Event Tree and Large Fault Tree SAIC's CAFTA Code	Database completed August 6, 1997.
3	CRYSTAL RIVER UNIT 3 Babcock & Wilcox Lowered-Loop Dry, Ambient Pressure Containment Region II	SAPHIRE 6.0 model only. CRP3- FTL Fault Tree Level 1 Load. At power conditions. ASP Category: PWR D	Florida Power Corporation's IPE (1993), plus updated data Modeling Methodology: Master Logic Diagram (Fault Tree Logic Model) SAIC's CAFTA Code	Database completed September 28, 1998.
4	D. C. COOK UNITS 1 & 2 Westinghouse Electric 4-Loop Wet, Ice-Condenser Containment Region III	SAPHIRE 6.0 model only. DC. COOK Full-Scope Level 1 Load: Linked Event Tree and Fault Tree Model At power conditions. ASP Category: PWR B	Indiana/Michigan Power Company IPE Model (1992), plus updated data Modeling Methodology: Small Event Tree and Large Fault Tree Westinghouse Electric's Code (e.g., GRAFTER, WLINK)	Database completed August 31, 1998.
5	FORT CALHOUN Combustion Engineering CE Dry, Ambient Pressure Containment Region IV	SAPHIRE 5.0 & 6.0 models. FORTCAL Full-Scope Level 1 Load: Linked Event Tree and Fault Tree Model At power conditions. ASP Category: PWR G	Omaha Public Power District's IPE (1993) Modeling Methodology: Small Event Tree and Large Fault Tree SAIC's CAFTA Code	Database completed November 22, 1996.

Idaho National Laboratory		Page: 24 of 77	
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236	
		Revision: 0	
		Effective Date: July 21, 2009	

6	FARLEY, JOSEPH M. UNITS 1 & 2 Westinghouse 3-Loop Dry, Ambient Pressure Containment Region II	SAPHIRE 5.0 & 6.0 models. FARLEY "Limited Full-Scope" Level 1 Load: Event Tree Model. Split- Fraction approach At power conditions. ASP Category: PWR B	Southern Nuclear Operating Company's PRA (1993) Modeling Methodology: Large Event Tree and Large Fault Tree Westinghouse Electric's Code: (e.g., GRAFTER, WLINK)	Fault tree models are not included in the database. Sequences can be generated at the "split-fraction" level only. "Limited full-scope" load completed January 9, 1994.
7	GRAND GULF UNIT 1 General Electric Type 6 Mark 3 Containment Region IV	SAPHIRE 5.0 NUREG-1150 model - GGULF-50 "Limited" Full-Scope Level 1 Load: Linked Event Tree and Fault Tree Model At power conditions. ASP Category: BWR C	NUREG/CR-4550, Volume 6, September 1989 Modeling Methodology: Small Event Tree and Large Fault Tree NUREG-1150 Model	This is a "Limited full-scope" load in that the nondominant sequences can be generated, but, no rules were created for adding recovery events. Database completed April 24, 1995.
8	OCONEE UNIT 3 Babcock & Wilcox Lowered-Loop Dry, Ambient Pressure Containment Region II	SAPHIRE 5.0 & 6.0 models. OCONE-FT Fault Tree Level 1 Load. At power conditions. ASP Category: PWR D	Duke Power Company's updated fault tree model (1997) Modeling Methodology: Master Logic Diagram (Fault Tree Logic Model) SAIC's CAFTA Code	Database completed May 7, 1997.
9	OCONEE STATION KEOWEE HYDRO STATION Region II	SAPHIRE 5.0 & 6.0 models. KEOWEE Fault Tree Level 1 Load.	Duke Power Company PRA (1990) Modeling Methodology: Master Logic Diagram (Fault Tree Logic Model) SAIC's CAFTA Code	Database completed December 28, 1995.
10	OYSTER CREEK General Electric Type 2 Mark 1 Containment Region I	SAPHIRE 5.0 & 6.0 models. OYCREEK "Limited" Full-Scope Level 1 Load: Dominant Accident Sequence Results Generated Using the Split- Fraction Approach At power conditions. ASP Category: BWR A	GPU Nuclear Corporation's IPE (1992) Modeling Methodology: Large Event Tree and Large Fault Tree Model PLG's RISKMAN Code	Sequences can be generated at the "split-fraction" level only. "Limited scope" database completed March 21, 1995.

Idaho National Laboratory		Page: 25 of 77	
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236	
	Revision:	0	
	Effective Date:	July 21, 2009	

11	PALO VERDE STATION Combustion Engineering CE80 Dry, Ambient Pressure Containment Region IV	SAPHIRE 5.0 & 6.0 models. PALVERDE & PVWIN-12 PALVERDE. Full-Scope Level 1 Load: DOS 5.0 Version of SAPHIRE PVWIN-12. Full-Scope Level 1 Load: Windows 6.0 Version of SAPHIRE Both Databases are Linked Event Tree and Fault Tree Model at power conditions.	Arizona Public Service Company's updated IPE Model (1994) Modeling Methodology: Small Event Tree and Large Fault Tree CRYSTAL Code (UNIX based)	PALVERDE database developed for use with DOS SAPHIRE. The PVWIN-12 database developed for use with Windows version of SAPHIRE; it included more cut sets (1E-12 truncation). PALVERDE database completed December 6, 1996. PVWIN-12 database completed December 22, 1996. Dominant and nondominant sequences can be generated with recovery events applied. External events (fire and seismic) can also be generated. PRA does not provide the data needed to adequately handle the flood analysis in the database. Thus, flood-related data are not included in the SAPHIRE database. Database completed October 30, 1997. Database completed October 19, 1994.
12	PEACH BOTTOM UNIT 2 General Electric Type 4 Mark 1 Containment Region I	SAPHIRE 6.0 NUREG-1150 model only. PBNUREG Full- Scope Level 1 Load: Linked Event Tree and Fault Tree Model At power conditions. ASP Category: BWR C	NUREG/CR-4550, Volume 4, August 1989 Modeling Methodology: Small Event Tree and Large Fault Tree NUREG-1150 Model	
13	RIVER BEND UNIT 1 General Electric Type 6 Mark 3 Containment Region IV	SAPHIRE 5.0 & 6.0 models. RIVERBEN Full-scope Level 1 load: Linked Event Tree and Fault Tree Model At power conditions. ASP Category: BWR C	Gulf States Utilities Company's IPE (1993) Modeling Methodology: Small Event Tree and Large Fault Tree. SAIC's CAFTA Code.	

Idaho National Laboratory		Page: 26 of 77	
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier:	INL/EXT-09-16236
		Revision:	0
		Effective Date:	July 21, 2009

14	SAN ONOFRE UNITS 2 & 3 Combustion Engineering CE Dry, Ambient Pressure Containment Region IV	SAPHIRE 5.0 & 6.0 models. SONGS2&3 Full-Scope Level 1 Load: Event Tree and Fault Tree Model At power conditions. ASP Category: PWR H	Southern California Edison Company's IPE (1993) Modeling Methodology: Cut Set Matching. ERIN Industries' REBECA Code.	Sequence cut sets are generated using previously generated system cut sets; i.e., the event trees are not directly linked to the fault trees. This is the same process used in the IPE. The database was completed June 6, 1994.
15	SEQUOYAH UNIT 1 Westinghouse Electric 4-Loop Wet, Ice-Condenser Containment Region II	SAPHIRE 5.0 & 6.0 models. SEQUOYAH "Limited" Full-Scope Level 1 Load: Linked Event Tree and Fault Tree Model. At power conditions. ASP Category: PWR B	NUREG/CR-4550, Volume 5, April 1990 Modeling Methodology: Small Event Tree and Large Fault Tree NUREG-1150 Model.	This is a "Limited" full-scope load in that the nondominant sequences can be generated, but, no rules were created for adding recovery events. The database was completed April 5, 1995.
16	SURRY UNIT 1 Westinghouse Electric 3-Loop Dry, Sub-Ambient Pressure Containment Region II	SAPHIRE 6.0 model. BNL-SDWN Full-Scope Level 1 Load: Linked Event Tree and Fault Tree Model At shutdown conditions ASP Category: PWR A1	Brookhaven National Laboratory's PRA (NUREG/CR-6144, June 1994) Modeling Methodology: Small Event Tree and Large Fault Tree NRC's SAPHIRE 5.0 Code	Initial database created by Brookhaven National Laboratory. INL modified the model to reduce the analysis time. BNL verified INL's database was functioning correctly. INL's version of database completed October 20, 1998.
17	SURRY STATION Westinghouse Electric 3-Loop Dry, Sub-Ambient Pressure Containment Region II	SAPHIRE 6.0 models only. SURRYLV1 and SURRYLV2 SURRYLV1. Full-scope Level 1 Load: SURRYLV2. Full-scope Level 2 Load: Both models are Linked Event Tree and Fault Tree Models At power conditions. ASP Category: PWR A1	Virginia Electric and Power Company's IPE updated model (1993) Modeling Methodology: Small Event Tree and Large Fault Tree NUS-Halliburton's NUPRA Code.	SURRYLV1 database completed July 14, 1998. SURRYLV2 database completed June 30, 1998.

Idaho National Laboratory		Page: 27 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

Table 3. SPAR Revision 2QA and LERF models available.

Item	SPAR Model Class	2QA Plant Name (Lead plant of each class in bold type)	Date Database Completed	LERF Model
1.	BWR A	Millstone 1	2/19/98	Available
2.	BWR A	Nine Mile Point 1	4/3/98	
3.	BWR A	Oyster Creek	7/14/97	
4.	BWR B	Dresden 2 and 3	7/14/97	
5.	BWR C	Browns Ferry 1, 2, and 3	5/1/98	
6.	BWR C	Brunswick 1 and 2	9/16/97	
7.	BWR C	Clinton 1	9/16/97	
8.	BWR C	Cooper Station	11/18/97	
9.	BWR C	Duane Arnold	11/18/97	
10.	BWR C	Enrico Fermi 2	12/1/97	
11.	BWR C	Fitzpatrick	12/1/97	
12.	BWR C	Grand Gulf 1	7/14/97	
13.	BWR C	Hatch 1 and 2	2/10/98	
14.	BWR C	Hope Creek 1	2/10/98	
15.	BWR C	LaSalle 1 and 2	7/14/97	Available
16.	BWR C	Limerick 1 and 2	3/3/98	
17.	BWR C	Monticello	1/14/98	
18.	BWR C	Nine Mile Pt. 2	1/14/98	
19.	BWR C	Peach Bottom 2 and 3	7/14/97	
20.	BWR C	Perry 1	7/14/97	
21.	BWR C	Pilgrim	1/14/98	
22.	BWR C	Quad Cities 1 and 2	3/23/98	
23.	BWR C	River Bend 1	3/16/98	
24.	BWR C	Susquehanna 1 and 2	2/19/98	
25.	BWR C	Vermont Yankee	2/19/98	Available
26.	BWR C	Columbia NP 2 (renamed from Washington)	2/10/98	
27.	PWR A1	Beaver Valley 1	2/2/98	
28.	PWR A1	North Anna 1 and 2	4/3/98	
29.	PWR A1	Surry 1 and 2	2/2/98	
30.	PWR A2	Beaver Valley 2	4/29/98	
31.	PWR A2	Millstone 3	5/1/98	
32.	PWR A2	South Texas 1 & 2	3/23/98	
33.	PWR B	Braidwood 1 and 2	2/2/98	
34.	PWR B	Byron 1 and 2	2/2/98	

**Software Acceptance Test Plan
for SAPHIRE Version 8 N6423**

Identifier: INL/EXT-09-16236
Revision: 0
Effective Date: July 21, 2009

Item	SPAR Model Class	2QA Plant Name (Lead plant of each class in bold type)	Date Database Completed	LERF Model
35.	PWR B	Callaway	2/2/98	
36.	PWR B	Catawba 1 and 2	3/3/98	
37.	PWR B	Comanche Peak 1 & 2	2/2/98	
38.	PWR B	Cook 1 and 2	4/16/98	
39.	PWR B	Diablo Canyon 1 and 2	4/16/98	
40.	PWR B	Farley 1 and 2	3/3/98	
41.	PWR B	Ginna	2/19/98	
42.	PWR B	Indian Point 2	4/29/98	
43.	PWR B	Kewaunee	4/16/98	
44.	PWR B	McGuire 1 and 2	3/23/98	
45.	PWR B	Pt. Beach 1 and 2	3/3/98	
46.	PWR B	Prairie Island 1 and 2	5/1/98	
47.	PWR B	Robinson 2	2/2/98	
48.	PWR B	Salem 1 and 2	2/10/98	
49.	PWR B	Seabrook 1	2/10/98	
50.	PWR B	Sequoyah 1 and 2	3/16/98	Available
51.	PWR B	Shearon Harris	3/23/98	
52.	PWR B	Summer 1	4/3/98	
53.	PWR B	Turkey Point 3 and 4	4/29/98	
54.	PWR B	Vogtle 1 and 2	3/16/98	
55.	PWR B	Watts Bar 1 and 2	3/23/98	
56.	PWR B	Wolf Creek 1	2/2/98	
57.	PWR D	Arkansas Nuclear One	4/29/98	
58.	PWR D	Crystal River 3	2/2/98	
59.	PWR D	Davis-Besse 1	5/1/98	
60.	PWR D	Oconee 1, 2, and 3	2/2/98	Available
61.	PWR D	Three Mile Island 1	2/19/98	
62.	PWR G	Calvert Cliffs 1 and 2	2/2/98	Available
63.	PWR G	Fort Calhoun 1	2/10/98	
64.	PWR G	Palisades	3/3/98	
65.	PWR G	St. Lucie 1	2/2/98	
66.	PWR G	St. Lucie 2	2/2/98	
67.	PWR G	Millstone 2	2/2/98	
68.	PWR H	Arkansas Nuclear One 2	4/29/98	
69.	PWR H	Palo Verde 1, 2, and 3	3/23/98	Available
70.	PWR H	San Onofre 2 and 3	2/19/98	
71.	PWR H	Waterford 3	3/3/98	

Idaho National Laboratory		Page: 29 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

Idaho National Laboratory		Page: 30 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

Table 4. SPAR Revision 3i models.

Item	SPAR Model Class	3-I Plant Model Name (Lead plant of each class in bold type)	Abbreviation	Date Database Completed
1.	PWR D	Arkansas Nuclear One 1	ANO1	May/19/2004
2.	PWR D	Arkansas Nuclear One 2	ANO2	Jun/09/2004
3.	BWR C	Browns Ferry 2	BFR2	Mar/11/2004
4.	BWR C	Browns Ferry 3	BFR3	Mar/11/2004
5.	BWR C	Brunswick 1 & 2	BRUN	May/19/2004
6.	PWR B	Braidwood 1 \$ 2	BRWD	May/19/2004
7.	PWR A1	Beaver Valley 1	BVS1	Mar/11/2004
8.	PWR A2	Beaver Valley 2	BVS2	Mar/11/2004
9.	PWR B	Byron 1 & 2	BYRN	May/19/2004
10.	PWR B	Callaway	CALL	May/19/2004
11.	PWR B	Catawaba 1 & 2	CATA	Jan/08/2004
12.	PWR G	Calvert Cliffs 1 & 2	CCLF	Jun/09/2004
13.	BWR C	Clinton	CLNT	Mar/11/2004
14.	PWR B	Cook 1 & 2	COOK	Feb/10/2004
15.	BWR C	Cooper Station	COOP	Jun/09/2004
16.	PWR B	Comanche Peak 1 & 2	COPK	Jun/09/2004
17.	PWR D	Crystal River	CRYS	May/06/2003
18.	PWR D	Davis-Besse 1	DAVB	Mar/18/2004
19.	PWR B	Diablo Canyon 1 & 2	DCAN	Jun/09/2004
20.	BWR B	Dresden 1 & 2	DRES	May/19/2004
21.	BWR C	Duane Arnold	ARND	May/19/2004
22.	PWR B	Farley	FARL	Jul/02/2003
23.	PWR G	Fort Calhoun	FCAL	Oct/18/2001
24.	BWR C	Enrico Fermi2	FERM	Mar/18/2004
25.	BWR C	Fitzpartick	FITZ	Mar/18/2004
26.	BWR C	Grand Gulf	GGUL	Aug/07/2002
27.	PWR B	Ginna	GINA	Jun/09/2004
28.	BWR C	Hatch	HATC	May/19/2004
29.	BWR C	Hope Creek	HOPE	Sep/10/2004
30.	PWR B	Indian Point 2	IPT2	Feb/10/2004
31.	PWR B	Indian Point 3	IPT3	Jun/02/2004
32.	PWR B	Kewanee	KEWA	Oct/27/2003
33.	BWR C	Limerick 1 & 2	LIMR	Jul/26/2004
34.	BWR C	LaSalle 1 & 2	LSAL	Feb/10/2004
35.	PWR B	McGuire 1 & 2	MCGU	Jan/08/2004
36.	PWR G	Millstone 2	MIL2	Apr/14/2004

Idaho National Laboratory		Page: 31 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

Item	SPAR Model Class	3-I Plant Model Name (Lead plant of each class in bold type)	Abbreviation	Date Database Completed
37.	PWR A2	Millstone 3	MIL3	May/24/2002
38.	BWR C	Monticello	MONT	Jan/08/2004
39.	PWR A1	North Anna	NANN	Nov/21/2003
40.	BWR A	Nine Mile Point 1	NMP1	Feb/10/2004
41.	BWR C	Nine Mile Point 2	NMP2	Jan/03/2003
42.	PWR D	Oconee Station 1, 2 & 3	OCON	Jun/29/2004
43.	BWR A	Oyster Creek	OYST	May/06/2003
44.	PWR G	Palisades	PALI	May/07/2004
45.	PWR B	Point Beach 1 & 2	PBCH	Jun/15/2004
46.	BWR C	Peach Bottom 2 & 3	PBOT	Oct/27/2003
47.	BWR C	Perry	PERRY	May/19/2004
48.	BWR C	Pilgrim	PILG	Nov/21/2003
49.	PWR B	Prairie Island	PRAI	Apr/14/2004
50.	PWR H	Palo Verde	PVNG	
51.	BWR C	Quad Cities 1 & 2	QCTY	Mar/22/2004
52.	BWR C	River Bend	RIVB	
53.	PWR B	Robinson 2	ROBN	Oct/02/2003
54.	PWR B	Salem	SALM	Sep/10/2004
55.	PWR B	Sea Brook	SBRK	Feb/10/2004
56.	PWR B	Sequoyah 1 & 2	SEQH	May/06/2003
57.	PWR H	San Onofre 2 & 3	SONG	Apr/14/2004
58.	PWR B	Shearon Harris	HARR	Oct/02/2003
59.	PWR A2	South Texas Project	STEX	Aug/07/2002
60.	PWR G	St. Lucie 1	STL1	Jun/02/2004
61.	PWR G	St. Lucie 2	STL2	Jun/02/2004
62.	PWR B	V. C. Summer 1	SUMM	May/19/2004
63.	PWR A1	Surry 1 & 2	SURY	Jun/29/2004
64.	BWR C	Susquehanna 1 & 2	SUSQ	Jan/03/2003
65.	PWR B	Turkey Point 3 & 4	TKPT	Jun/15/2004
66.	PWR D	Three Mile Island 1	TMI1	Jan/03/2003
67.	PWR B	Vogtle 1 and 2	VBOGT	Jan/04/2008
68.	BWR C	Vermont Yankee	VYAN	May/19/2004
69.	PWR B	Watts Bar 1 and 2	WBAR	May/06/2003
70.	BWR C	Columbia Nuclear Power 2	CNPS	
71.	PWR B	Wolf Creek	WOLF	May/19/2004
72.	PWR H	Waterford	WTRF	May/19/2004

Idaho National Laboratory		Page: 32 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

In addition, three theoretical plant models were created to test features not already represented in actual plants. These plants are “DEMO,” “TSTU,” and “SIMPLE-FT.” These plants: “TSTU” and “SIMPLE-FT” are used internally for testing purposes only and are not available for distribution. The “DEMO” plant is delivered with the installation package and is used to verify correct installation and also used in training exercises

A set of 72 simplified plant analysis risk (SPAR) models (Revision 2QA) for the NRC's Accident Precursor Program. This is a set of reasonably accurate, consistent, and representative Level 1 PRA models for operating plants for use in evaluating operational event analysis (Long et al. 1998).

Several classes of plants were identified within the 72 SPAR models: four classes of boiling water reactors (BWRs) and six classes of pressurized water reactors (PWRs), based on similar plant responses to transients and accidents and the systems designed to perform those responses. The lead plant model was chosen for each class as the testing tool for the verification and validation effort. Models other than the lead plant within a particular class could have been used as additional tests, but we expected that, since additional models are derivatives from the lead plant in its category, the potential for capturing a SAPHIRE problem not already found in the lead-plant model is low.

Originally, the set of written procedures developed for manually testing the ten lead plant models were obtained. These procedures were translated into an automated test script, initially for the Byron (BYRN) and Peach Bottom (PBOT) models. Once these were completed, the automated test scripts for the other lead plant models were easily developed by changing only the inputs and the result tables associated with the script. These changes were subsequently checked to ensure that the proper analysis steps were being applied.

In general, SPAR model testing focuses on solving fault trees, solving accident sequences from the event trees, and using the Events and Condition Assessment and Initiating Events Assessment to apply change sets for sensitivity analyses. The other SPAR models in the overall test suite are Dresden (DRES), Grand Gulf (GGUL), Milstone (MIL3), Oconee (OCON), Oyster Creek (OYST), San Onofre (SONG), St Lucie (STL1), and Surry (SURY).

3.2.1.1 Plant Model SUR40

A limited study was performed using a Surry Level 1, NUREG-1150 database developed with SAPHIRE 4.0 to test SAPHIRE versions through SAPHIRE Version 6.42 (McCabe 1998). No changes were made to the original Surry database other than those necessary to upgrade the data to the current versions. Results of all 45 accident sequences were recorded and tracked through a series of SAPHIRE releases. The differences from version to version, if any, are documented and explained. Results tracked included minimal cut set upper bound (i.e., min-cut upper bound) and the number of cut sets per sequence. In addition to these results, the PRA database overall uncertainty, consisting of the mean, 5th, and 95th percentile results, were recorded. For this testing, the latest results of the study were used as a baseline for the SUR40 database. The analysis method for generating the values to be compared was translated into an automated test script. Then, this test was added to the overall test suite.

3.2.1.2 Plant Model TSTU

Idaho National Laboratory		Page: 33 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

PRA analysts developed most models chosen for testing to represent real models and data. An exception is a model we refer to as TSTU, a database identifier for Test of Uncertainty. For uncertainty testing, we developed a simple database for testing the spectrum of different distributions. This test database contains a series of simple fault tree models, each consisting of a single OR gate with a single basic event as its input. These basic events were assigned a different distribution. Use of this simple database allowed the test developer and the statistician to verify correct results of the uncertainty tests.

3.2.1.3 Plant Model DEMO

The DEMO database, packaged with SAPHIRE, is a simple model consisting of a single event tree, two fault trees, and 24 basic events. It was designed to demonstrate the basic features of SAPHIRE. Although small, the model and its data are manipulated in the same way as other models. For this reason, and in the interests of speed, the DEMO project was deemed sufficient, as in earlier verification and validation, for testing importance measures and as one of the models for testing change sets.

Importance measures are calculated for individual events found in a list of cut sets. Since the history of the cut sets are immaterial at the point in a PRA study where importance measures are generated, it was sufficient to use DEMO cut sets to perform importance measures testing. This database provides a succinct set of minimal cut sets from which to derive the importance measures.

Change sets, a SAPHIRE mechanism for performing sensitivity analysis and which operate only on basic event data, were tested using the DEMO database. The function of a change set is to temporarily modify one or more basic event's data characteristics (known as the current event data). All cut set operations exclusively use the current event data, regardless of database size and complexity. The mechanism for creating current event data with basic event changes is the same, regardless of which model is used.

3.2.1.4 Plant Model SURRY-50

Surry-50 is a Level 1, NUREG-1150 database developed with SAPHIRE 5.0. It was adopted for use in testing change sets and generating and quantifying cut sets. These analysis options included sequences with flag sets, the application of recovery rules, and cut set updating (e.g., re-quantifying cut sets without regenerating them from scratch).

The Surry PRA involves a two-stage event tree analysis process. The first stage uses a set of event trees to develop the dominant accident event sequences that lead to core damage. The second stage uses a different set of event trees, identified in the PRA as bridge event trees, to aggregate the core damage sequences into plant damage states. The SAPHIRE model developers performed quality assurance checks of the data to ensure completeness and accuracy of the data input to the database. The dominant accident sequences in the database were benchmarked against the results reported in the source PRA, and the principal author of the original PRA reviewed them.

3.2.1.5 Plant Model COM-PEAK

Comanche Peak is a Level 1 Individual Plant Examination (IPE) database developed with SAPHIRE 6.0. It, like SURRY-50, is used for testing cut set generation and quantification, recovery rules, and cut set updating. It is included for additional cut set verification and for testing fault tree flag sets.

Idaho National Laboratory		Page: 34 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

3.2.1.6 Plant Model BV2-5

Beaver Valley 2, Version 5, is a Level 1 PRA developed from the plant's IPE, using SAPHIRE 5. This database was specifically constructed as a train-level database, since modeling is simplified to the system train level. The model defines 17 plant damage states (end states) associated with its Level 1 event trees. Plant damage states are defined by the Reactor Control System (RCS) pressure, availability of containment heat removal systems, and status of containment isolation or bypass at the time of reactor vessel failure after core damage. Experienced PRA analysts recommended this model as a good database for end state analysis testing, particularly for importance measures and uncertainty testing.

3.2.1.7 Plant Model CR3

Crystal River 3, Version 6, is a Level 1 PRA developed from the plant's IPE, using SAPHIRE 6. This is a large fault tree model based on the master logic diagram modeling methodology. The master logic diagram method translates all PRA accident sequences into a fault tree (where each sequence is input into a top OR gate). Then, this single fault tree is further developed until the entire PRA is represented within the fault tree. Experienced PRA analysts recommended this model as a good Version 6 database for data manipulation and fault tree paging tests. A unique feature of this model is that since the PRA logic is contained with a single fault tree, the size of the fault tree is somewhat large.

3.2.1.8 Plant Model S_LERF

Surry large early release frequency (LERF) is a Level 2 SPAR model. It offers a good working example of sequence and end state analysis. In particular, it links event trees using both small and large event tree methods. Further, it partitions and gathers cut sets by accident sequence using the partition rules option built into SAPHIRE.

Since this model is a Level 1/Level 2 model, the S_LERF model links the Surry Level 1 event trees (used to calculate core damage frequency) to the Surry Level 2 event tree (used for calculating LERF). This link is performed by a bridge event tree that generates plant damage states. These plant damage states are then used as initiating events to the LERF event trees. The plant damage states are generated by querying the Level 1 bridge event tree sequence logic to determine the seven to eight characteristics of the plant damage state using "IF-THEN-ELSE" partition rules. All cut sets associated with each Level 1 bridge event tree sequence are assigned to the same plant damage state using the "GlobalPartition" feature of SAPHIRE. In addition, plant damage states that have been determined to produce identical LERF results have been collapsed into a representative plant damage state using the "CurrentPart" feature of SAPHIRE.

3.3 Testing Approach

The test approach used for SAPHIRE Version 8 will be based upon the test approach for previous releases of the SAPHIRE tool. Taking into consideration lessons learned from the SV&V efforts, where applicable, actual tests and test specifications from the older testing were used in the SV&V. Additional tests were developed specifically for the newer SV&V process, primarily due to the fact that the test could be automated. This automation aspect of testing allows the testing team to rerun a battery of calculations as many times as they wish, regardless of the complexity of the test. In order to decide which tests were to be used and why, a test plan was developed. This plan followed the general procedures used in the earlier

Idaho National Laboratory		Page: 35 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

V&V efforts, but was modified to take advantage of unique features found when performing automated testing. Thus, the updated testing plan for the SV&V includes the following steps:

- Preparing an SV&V plan.
- Determining the areas requiring testing. This step is similar to the V&V process of identifying vital and non-vital functions. Additional features are checked in the current V&V process then were tested in the previous V&V efforts.
- Developing the test model, including the identification of available SAPHIRE PRA databases that would adequately test SAPHIRE functions.

Availability of a variety of different plant data models enhances the viability of the test suite. Core features are exercised repeatedly across tests (and their associated models) in the process of performing each test's specific analysis task. Use of different plant and database models, from the simple DEMO database to the SPAR Revision 2QA, SPAR 3i models to NUREG-1150 models, provides quality and reliability assurance that any variations among models are appropriately handled by any released version of SAPHIRE. While the current tests do not address every feature within SAPHIRE, they do cover the important calculational parts of the software. Also, some specific PRA areas are tested using only a single test case.

Models and test cases are added as needed to the test suite to improve the overall coverage of testing vital functions in SAPHIRE.

3.3.1 Test Descriptions

This section contains a complete list of test descriptions, referenced by one or more test scenarios in the report. Note that, unless otherwise specified, SAPHIRE Version 8 test results were compared with results from SAPHIRE Version 7 test results. The plant models used for the tests are listed in the two tables at the end of this section, Table 6.2.1-1. and Table 6.2.1-2., which provide a cross reference of the tests, the plants used, and the specific features tested in each scenario. Each test may be performed on multiple plant models.

Cut set probability cutoff values used in generating fault tree and sequences cut sets are 1E-15 for BWRs and 1E-16 for PWRs.

3.3.1.1 Test-01, Solve Fault Trees

Scenarios generate basic event data (with no change sets), solve (with cut set probability cutoff) and quantify fault tree minimal cut sets, and recovery rules. The alternate case min cut upper bound, base case min cut upper bound, and cut set totals are verified for each fault tree

3.3.1.2 Test-02, CoreDamage Frequency

Idaho National Laboratory		Page: 36 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

Scenarios generate basic event data (with no change sets), solve (with cut set probability cutoff) and quantify sequence minimal cut sets, and recovery rules. The alternate case min cut upper bound, base case min cut upper bound, and cut set totals are verified for each sequence.

3.3.1.3 Test-03, Events and Condition Assessment: Auxiliary Feed Water out of service for 72 hours

Scenarios exercise all aspects of operational event analysis including removal of equipment from service and automated processing of all steps. These steps include basic event generation with change sets; and generation, quantification, and recovery of cut sets. The number of sequences; total CCDP; total core damage probability (CDP); total importance; and CCDP, CDP, and importance for each sequence are verified.

3.3.1.4 Test-04, Events and Condition Assessment: Emergency Diesel Generator out of service for three months

Scenarios exercise all aspects of operational event analysis, including removal of equipment from service and automated processing of all steps. These steps include basic event generation with change sets, and generation, quantification, and recovery of cut sets. The number of sequences, total CCDP, total core damage probability (CDP), total importance, and CCDP, CDP, and importance for each sequence are verified.

3.3.1.5 Test-05, Initiating Event Assessment: Transient with no other failures

Scenarios exercise the number of sequences; total CCDP; total core damage probability (CDP); total importance; and CCDP, CDP, and importance for each sequence are verified. Automated steps performed for initiating event assessments include basic event generation with change sets; and generation, quantification, and recovery of cut sets.

3.3.1.6 Test-06, Initiating Event Assessment: Small Loss of Coolant Accident (SLOCA) with no other failures

Scenarios exercise the number of sequences; total CCDP; total core damage probability (CDP); total importance; and CCDP, CDP, and importance for each sequence are verified. Automated steps performed for initiating event assessments include basic event generation with change sets; and generation, quantification, and recovery of cut sets

3.3.1.7 Test-07, Initiating Event Assessment: Steam Generator Tube Rupture with no other failures

Scenarios exercise the number of sequences; total CCDP; total core damage probability (CDP); total importance; and CCDP, CDP, and importance for each sequence are verified. Automated steps performed for initiating event assessments include basic event generation with change sets; and generation, quantification, and recovery of cut sets

3.3.1.8 Test-08, Initiating Event Assessment: Grid-Related Loss of Off-Site Power (LOOP) with no other failures

Idaho National Laboratory		Page: 37 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

Scenarios exercise the number of sequences; total CCDP; total core damage probability (CDP); total importance; and CCDP, CDP, and importance for each sequence are verified. Automated steps performed for initiating event assessments include basic event generation with change sets; and generation, quantification, and recovery of cut sets.

3.3.1.9 Test-09, Initiating Event Assessment: Plant-Centered LOOP with no other failures

Scenarios exercise the number of sequences; total CCDP; total core damage probability (CDP); total importance; and CCDP, CDP, and importance for each sequence are verified. Automated steps performed for initiating event assessments include basic event generation with change sets; and generation, quantification, and recovery of cut sets.

3.3.1.10 Test-10, Initiating Event Assessment: Severe Weather LOOP with no other failures

Scenarios exercise the number of sequences; total CCDP; total core damage probability (CDP); total importance; and CCDP, CDP, and importance for each sequence are verified. Automated steps performed for initiating event assessments include basic event generation with change sets; and generation, quantification, and recovery of cut sets.

3.3.1.11 Test-11, Initiating Event Assessment: Extreme Severe Weather LOOP with no other failures

Scenarios exercise the number of sequences; total CCDP; total core damage probability (CDP); total importance; and CCDP, CDP, and importance for each sequence are verified. Automated steps performed for initiating event assessments include basic event generation with change sets; and generation, quantification, and recovery of cut sets.

3.3.1.12 Test-12, Initiating Event Assessment: Transient with AFW Failed

Scenarios exercise the number of sequences; total CCDP; total core damage probability (CDP); total importance; and CCDP, CDP, and importance for each sequence are verified. Automated steps performed for initiating event assessments include basic event generation with change sets; and generation, quantification, and recovery of cut sets.

3.3.1.13 Test-13, Dominant Sequence Frequencies and Core Damage Frequency Uncertainty

This scenario continues the tracking with an automated test script. Cut sets generated with cut set probability cutoff and cut set size cutoff. Recovery rules are applied without cutoff. Cut set update performed with no truncation. Project level Monte Carlo uncertainty performed on results using 5000 samples.

3.3.1.14 Test-14, Fault Tree Uncertainty: Monte Carlo Method/Log Normal Distribution

This scenario consists of six variations that test uncertainty using the Monte Carlo simulation technique for the log normal distribution type. The six variations use fault trees that consists of an OR

Idaho National Laboratory		Page: 38 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

gate with a single basic event as its input. Each variation uses differing basic event nominal probabilities and error factors. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples (simulated values) and a random number seed of 4,321 for each test.

3.3.1.15 Test-15, Fault Tree Uncertainty: Monte Carlo Method/Normal Distribution

This scenario consists of variations that test uncertainty using the Monte Carlo simulation technique for the normal distribution type. Two fault trees are used that consist of an OR gate with a single basic event as its input, with differing basic event nominal probabilities and standard deviation values. Fault tree combinations of five sample sizes and two seed values are used for a total of ten tests for each tree. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified.

3.3.1.16 Test-16, Fault Tree Uncertainty: Monte Carlo Method/Beta Distribution

This scenario consists of ten variations that test uncertainty using the Monte Carlo simulation technique for the beta distribution type. The ten variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event nominal probabilities and uncertainty values. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples and a seed of 4,321 for each test.

3.3.1.17 Test-17, Fault Tree Uncertainty: Monte Carlo Method/Chi Squared Distribution

This scenario consists of twelve variations that test uncertainty using the Monte Carlo simulation technique for the chi-square distribution type. For ten of the variations, ten fault trees are used that consists of an OR gate with a single basic event as its input. Each basic event has a different nominal probability and uncertainty value (degrees of freedom). The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples and a seed of 4,321 for each test. For the other variations two fault trees are used that consist of an OR gate with a single basic event as its input with differing basic event nominal probabilities and uncertainty values. For each of these fault trees, four different sample sizes and seed of 4,321 are used. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified.

3.3.1.18 Test-18, Fault Tree Uncertainty: Monte Carlo Method/Exponential Distribution

This scenario consists of eight variations that test uncertainty using the Monte Carlo simulation technique for the exponential distribution type. The eight variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event nominal probabilities. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples and a seed of 4,321 for each test.

3.3.1.19 Test-19, Fault Tree Uncertainty: Monte Carlo Method/Uniform Distribution

This scenario consists of four variations that test uncertainty using the Monte Carlo simulation technique for the uniform distribution type. The four variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event nominal probabilities and upper end uncertainty values. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples and a seed of 4,321 for each test.

Idaho National Laboratory		Page: 39 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

3.3.1.20 Test-20, Fault Tree Uncertainty: Monte Carlo Method/Gamma Distribution

This scenario consists of six variations that test uncertainty using the Monte Carlo simulation technique for the gamma distribution type. The six variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event nominal probabilities and uncertainty values (r). The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples and a seed of 4,321 for each test.

3.3.1.21 Test-21, Fault Tree Uncertainty: Monte Carlo Method/Maximum Entropy Distribution

This scenario consists of seven variations that test uncertainty using the Monte Carlo simulation technique for the maximum entropy distribution type. The seven variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event nominal probabilities and upper end and lower end uncertainty values. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples and a seed of 4,321 for each test.

3.3.1.22 Test-22, Sequence Uncertainty: Monte Carlo Method / Dirichlet Distribution

This test scenario consists of four variations that test uncertainty analyses using the Monte Carlo simulation technique for the Dirichlet distribution type. The first three variations each use a three-branch event tree with differing failure probabilities and parameter values. The fourth variation uses a 121-branch event tree. Change sets are used to correlate the basic events. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified.

3.3.1.23 Test-23, Fault Tree Uncertainty: Monte Carlo Method/Seismic Distribution

This scenario consists of four variations that test uncertainty using the Monte Carlo simulation technique for the seismic distribution type. The four variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event median failure acceleration, screening G-level, Beta-R and Beta-U values. Uncertainty analysis is performed using the Seismic analysis type. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 10,000 samples and a seed of 4,321 for each test.

3.3.1.24 Test-24, Fault Tree and Sequence Uncertainty: Monte Carlo Method/Constrained Noninformative Distribution

This scenario consists of five variations that test uncertainty using the Monte Carlo simulation techniques for the Constrained Noninformative distribution type. The three variations involving fault trees use fault trees that consists of an OR gate with a single basic event as its input with differing basic event nominal probabilities. The two variations involving sequences use event trees with differing initiating event nominal frequencies. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 10,000 simulated values for each test.

3.3.1.25 Test-25, Fault Tree Uncertainty: Latin Hypercube Method/Log Normal Distribution

Idaho National Laboratory		Page: 40 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

This scenario consists of six variations that test uncertainty using the Latin Hypercube simulation technique for the log normal distribution type. The six variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event nominal probabilities and error factors. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples (simulated values) and a random number seed of 4,321 for each test.

3.3.1.26 Test-26, Fault Tree Uncertainty: Latin Hypercube Method/Normal Distribution

This scenario consists of variations that test uncertainty using the Latin Hypercube simulation technique for the normal distribution type. Two fault trees are used that consist of an OR gate with a single basic event as its input, with differing basic event nominal probabilities and standard deviation values. Fault tree combinations of five sample sizes and two seed values are used for a total of ten tests for each tree. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified.

3.3.1.27 Test-27, Fault Tree Uncertainty: Latin Hypercube Method/Beta Distribution

This scenario consists of ten variations that test uncertainty using the Monte Carlo simulation technique for the beta distribution type. The ten variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event nominal probabilities and uncertainty values. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples and a seed of 4,321 for each test.

3.3.1.28 Test-28, Fault Tree Uncertainty: Latin Hypercube Method/Chi Squared Distribution

This scenario consists of twelve variations that test uncertainty using the Monte Carlo simulation technique for the chi-square distribution type. For ten of the variations, ten fault trees are used that consists of an OR gate with a single basic event as its input. Each basic event has a different nominal probability and uncertainty value (degrees of freedom). The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples and a seed of 4,321 for each test. For the other variations two fault trees are used that consist of an OR gate with a single basic event as its input with differing basic event nominal probabilities and uncertainty values. For each of these fault trees, four different sample sizes and seed of 4,321 are used. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified.

3.3.1.29 Test-29, Fault Tree Uncertainty: Latin Hypercube Method/Exponential Distribution

This scenario consists of eight variations that test uncertainty using the Monte Carlo simulation technique for the exponential distribution type. The eight variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event nominal probabilities. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples and a seed of 4,321 for each test.

3.3.1.30 Test-30, Fault Tree Uncertainty: Latin Hypercube Method/Uniform Distribution

Idaho National Laboratory		Page: 41 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

This scenario consists of four variations that test uncertainty using the Monte Carlo simulation technique for the uniform distribution type. The four variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event nominal probabilities and upper end uncertainty values. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples and a seed of 4,321 for each test.

3.3.1.31 Test-31, Fault Tree Uncertainty: Latin Hypercube Method/Gamma Distribution

This scenario consists of six variations that test uncertainty using the Monte Carlo simulation technique for the gamma distribution type. The six variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event nominal probabilities and uncertainty values (r). The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples and a seed of 4,321 for each test.

3.3.1.32 Test-32, Sequence Uncertainty: Latin Hypercube Method/Maximum Entropy Distribution

This scenario consists of seven variations that test uncertainty using the Monte Carlo simulation technique for the maximum entropy distribution type. The seven variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event nominal probabilities and upper end and lower end uncertainty values. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples and a seed of 4,321 for each test.

3.3.1.33 Test-33, Sequence Uncertainty: Latin Hypercube Method/Dirichlet Distribution

This test scenario consists of four variations that test uncertainty analyses using the Monte Carlo simulation technique for the Dirichlet distribution type. The first three variations each use a three-branch event tree with differing failure probabilities and parameter values. The fourth variation uses a 121-branch event tree. Change sets are used to correlate the basic events. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified. Since this distribution type was not available in version 5, version 6 results have been inspected for acceptance and are used for comparison against subsequent incremental releases.

3.3.1.34 Test-34, Fault Tree Uncertainty: Latin Hypercube Method/Seismic Distribution

This scenario consists of four variations that test uncertainty using the Monte Carlo simulation technique for the seismic distribution type. The four variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event median failure acceleration, screening G-level, Beta-R and Beta-U values. Uncertainty analysis is performed using the Seismic analysis type. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 10,000 samples and a seed of 4,321 for each test.

3.3.1.35 Test-35, Fault Tree and Sequence Uncertainty: Latin Hypercube Method/Constrained Noninformative Distribution

This scenario consists of five variations that test uncertainty using the Monte Carlo simulation techniques for the Constrained Noninformative distribution type. The three variations involving fault trees

Idaho National Laboratory		Page: 42 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

use fault trees that consists of an OR gate with a single basic event as its input with differing basic event nominal probabilities. The two variations involving sequences use event trees with differing initiating event nominal frequencies. The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 10,000 simulated values for each test.

3.3.1.36 Test-36, Fault Tree Uncertainty: Monte Carlo Method/Histogram Distribution

This scenario consists of four variations that test uncertainty using the Monte Carlo simulation technique for the histogram distribution type. The four variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event nominal probabilities and histograms (of percentage, area, and range types). The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples and a seed of 4,321 for each test.

3.3.1.37 Test-37, Fault Tree Uncertainty: Latin Hypercube Method/Histogram Distribution

This scenario consists of four variations that test uncertainty using the Latin Hypercube simulation technique for the histogram distribution type. The four variations use fault trees that consists of an OR gate with a single basic event as its input. Each variation uses differing basic event nominal probabilities and histograms (of percentage, area, and range types). The 5th percentile, 50th percentile, 95th percentile, and standard deviation results are verified based on 5,000 samples and a seed of 4,321 for each test.

3.3.1.38 Test-38, Gathering of End States

This scenario generates basic event data (with no change sets) and gathers the end states (without cut set probability cutoff, by sequence end state). The alternate case min-cut upper bound and the number of cut sets are verified for each end state.

3.3.1.39 Test-39, End State Uncertainty: Monte Carlo Method

These scenarios perform multiple event sampling on all sequences that belong to a particular end state (single uncertainty), as well as the collection of all end states (group uncertainty). The mean, 5th percentile, median, 95th percentile, and standard deviation results are verified based on 3,000 simulated values for each test.

3.3.1.40 Test-40, End State Uncertainty: Latin Hypercube Method

These scenarios perform multiple event sampling on all sequences that belong to a particular end state (single uncertainty), as well as the collection of all end states (group uncertainty). The mean, 5th percentile, median, 95th percentile, and standard deviation results are verified based on 3,000 simulated values for each test.

3.3.1.41 Test-41, Cut Set Verification

This test case consists of scenarios that compare cut sets from selected fault trees, sequences, and end states. The cut set frequency, percent contribution to the total, and basic events in the cut set are verified. Cut sets are solved and /or /gathered with truncation, auto-recovered, and updated. Sequences and fault trees are solved with and without their default flag sets. Also, fault tree editing is briefly tested. This is done by opening the alphanumeric logic editor, saving and converting logic to graphics, then

Idaho National Laboratory		Page: 43 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

pulling up the graphical editor and saving the graphics. This test does not test specific editing features but it does verify that the original logic is correctly loaded and saved. Failure of the logic to be preserved correctly would be detected with incorrect cut set results

3.3.1.42 Test-42, Link Small Event Tree

This scenario uses the Surry Large Early Release Frequency (LERF) Level 2/3 model (S_LERF) to link event trees using the small event tree methodology. Prior to link, each event tree is loaded into the graphical editor and saved to ensure that the correct logic is preserved. The sequences are then solved with cutoff. The alternate case min cut upper bound and number of cut sets is verified for each Level 1 sequence.

3.3.1.43 Test-43, Partition Sequence Cut Sets

This scenario applies event tree partition rules to the sequences generated in scenario reference number Test-42. These partition rules assign Plant Damage States (PDSs) to all sequences with cut sets. These end states are then gathered by cut set partition. The alternate case min cut upper bound and number of cut sets is verified for each PDS.

3.3.1.44 Test-44, Link Large Event Tree

This scenario uses the results from scenario reference number Test-43. The PDS event trees created by the partition rules are linked using the large event tree methodology and create sequence logic cut sets. The LERF end states are then gathered by sequence end state and re-quantified using the Rare Event approximation. The alternate case min-cut upper bound and number of cut sets are verified for each LERF end state.

3.3.1.45 Test-45, Fault Tree Importance Measures

This test case consists of scenarios that test importance measures calculations with fault trees for each of the importance measures: ratio, difference, and uncertainty. For each event, the name, number of occurrences, probability, Fussell-Vesely (or Birnbaum or uncertainty importance), risk reduction ratio (or difference), risk increase ratio (or difference) results are verified.

3.3.1.46 Test-46, Sequence Importance Measures

This test case consists of scenarios that test Sequence importance measures calculations for each of the importance measures: ratio, difference, and uncertainty. For each event, the name, number of occurrences, probability, Fussell-Vesely (or Birnbaum or uncertainty importance), risk reduction ratio (or difference), risk increase ratio (or difference) results are verified.

3.3.1.47 Test-47, Sequence Group Importance Measures

This test case consists of scenarios that test Sequence Group importance measures calculations for each of the importance measures: ratio, difference, and uncertainty. For each event, the name, number of occurrences, probability, Fussell-Vesely (or Birnbaum or uncertainty importance), risk reduction ratio (or difference), risk increase ratio (or difference) results are verified.

3.3.1.48 Test-48, End State Importance Measures

Idaho National Laboratory		Page: 44 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

This test case consists of scenarios that test End State importance measure calculations for each of the importance measures: ratio, difference, and uncertainty. For each event, the name, number of occurrences, probability, Fussell-Vesely (or Birnbaum or uncertainty importance), risk reduction ratio (or difference), risk increase ratio (or difference) results are verified.

3.3.1.49 Test-49, End State Group Importance

This test case consists of scenarios that test End State Group importance measures calculations for each of the importance measures: ratio, difference, and uncertainty. For each event, the name, number of occurrences, probability, Fussell-Vesely (or Birnbaum or uncertainty importance), risk reduction ratio (or difference), risk increase ratio (or difference) results are verified.

3.3.1.50 Test-50, Change Set Processing: Single

This test case consists of scenarios that test the effects of basic event changes, via change sets, on sequence cut set results. In these scenarios, single basic event changes are made in a change set. The change set is then marked and the basic event data is generated. An affected sequence is then selected and cut set results are verified.

3.3.1.51 Test-51 Change Set Processing: Class

This test case consists of scenarios that test the effects of basic event changes, via change sets, on sequence cut set results. In these scenarios, class basic event changes are made in a change set. The change set is then marked and the basic event data is generated. An affected sequence is then selected and cut set results are verified.

3.3.1.52 Test-52, Change Set Processing: Marked Order

This test case consists of scenarios that test the effects of basic event changes, via change sets, on sequence cut set results. In these scenarios, the change sets created in Test-50 and Test-51 are used. Multiple change sets are marked and the basic event data is generated. An affected sequence is then selected and cut set results are validated. This test verifies that the changed basic events are processed correctly based on the marked order of the change sets.

3.3.1.53 Test-53, Extract, Delete, Load, Solve: Fault Trees and Basic Events

This test consists of scenarios that exercise utility functions associated with the database for loading plant models, end state data or other information to be analyzed with the tool set.

3.3.1.54 Test-54, Fault Tree Utility Functions: Auto page, Solve, Save Cut Sets to End States

SAPHIRE provides several utilities maintain fault trees. These tests verify that the use of these features does not introduce errors into the database. The auto-page scenario breaks up a large fault tree into manageable smaller fault trees with transfer information. An auto-page is performed on a large fault tree, and then the modified tree is solved to verify the cut set results are not altered with the paging operation. Another scenario copies a fault tree cut sets to an end state, and then verifies that the cut sets in the end state match the cut sets in the fault tree.

Idaho National Laboratory		Page: 45 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

3.3.1.55 Test-55, Event Tree Linking (including rules)

The event tree linking rules are tested using several different databases. The databases are the Surry LERF model, Wolf Creek Revision 302, and Peach Bottom Revision 302. The Surry LERF model links the Level 1 event tree sequences together prior to solving the accident sequences, then performs an end state gather. The end states then become Level 2 event trees, which are linked together using the large event tree method. These Level 2 sequences are then gathered into the final end states for LERF, NO-LERF, etc. The Wolf Creek and Peach Bottom models have no accident sequences at the beginning. The test has the sequences being generated using dynamic flag sets for the accident sequences, and then evaluates the sequences. The sequences are evaluated using the developed dynamic flag sets and then with no flag sets.

3.3.1.56 Test-56, End-State Gathering

The end state gathering process is tested using the Surry LERF model and the Beaver Valley NUREG 1150 model. Both models have the sequences gathered into end states. The Surry LERF model uses partition rules, while the Beaver Valley model uses the end state name.

3.3.1.57 Test-57, Compound Event Plug-ins

The compound event plug-in is being tested for both the common cause module, utility module (i.e. add, multiply), and load-capacity. The scenarios include testing the utility module and load-capacity, testing the add and multiply functions in order to calculate the probability of the compound event. Then change sets are created to affect the compound event and the final probability. The results are verified to make sure the probability is correct. Also tested is the load-capacity plug-in. The values are input and the probability is calculated along with performing an uncertainty calculation. The input values are also modified using a change set and then a new probability along with uncertainty evaluation is performed and verified to be correct.

3.3.1.58 Test-58, Base Case Update

All models have fault tree results and accident sequences cut sets converted to the base case. This scenario is for fault tree cut sets converted to the base case for comparison to the current case using change sets.

3.3.1.59 Test-59, Calculation Types

The calculation types are tested. The “TRUE” calculation type is tested. The “TRUE, FALSE, and IGNORE” calculation types are tested. Fault trees are developed to verify the different calculation types are being changed in the change sets and the results are correct. The other calculation types (i.e., 3, 5, and 7) are also being checked in the simple database using change sets.

3.3.1.60 Test-60 Application of Change Sets

Change sets are used in numerous databases. Both class and single event change sets are developed and tested. The change sets test both probability changes and calculation type changes.

3.3.1.61 Test-61, Uncertainty Distributions

Idaho National Laboratory		Page: 46 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

All of the uncertainty distribution types are tested.

3.3.1.62 Test-62, N of M Gates

The N/M gates are tested using the simple database (SIMPLE-FT) plant model. The N/M gate has multiple N/M gates feeding into it. The N/M gate is evaluated using all of the inputs and also with inputs affected by change sets.

3.3.1.63 Test-63, Sequence Stress Testing

Several scenarios test sequence stress (i.e., numerous sequences being generated). An event tree links over and over in order to test the ability to generate numerous sequences correctly.

3.3.1.64 Test-64, Calculations on the Common-Cause Plug-in

Use of the common-cause plug-ins is verified. Basic events are tested by using change sets. One set of the inputs is set TRUE. This requires SAPHIRE to re-calculate the Common Cause Failure (CCF) plug-in basic event for evaluation. The final probability is manually calculated and checked to the probability calculated for final verification.

3.3.1.65 Test-65, Event Transformations

This test is currently in the concept design phase.

3.3.1.66 Test-66, Wrong Results

This test verifies the output of results. The output from the test is compared to know incorrect results to verify that the comparison is worked correctly. A LOSP scenario is executed for comparison.

Table of Features Tested by the Automated Test Suite

Table 1 lists the primary areas addressed for the test suite, where the applicable model and test number are identified for each area.

The top row indicates sets of test scenario identifiers. The test scenario identifiers consist of an abbreviation of the plant model tested and a number unique to that plant. After the plant model abbreviation and number, test identifier numbers are noted in parentheses. These tests were previously explained in this same section. The left-most column of Table 6-4 lists the features tested. An "X" in the column indicates the features the set (column) of scenarios tested and which plants they were tested with. A blank indicates that the set of test scenarios does not use the feature on that row as a primary test metric

Idaho National Laboratory			Page: 47 of 77			
Software Acceptance Test Plan for SAPHIRE Version 8 N6423			Identifier:	INL/EXT-09-16236		
			Revision:	0		
			Effective Date:	July 21, 2009		

Table 5. Features tested by the automated test suite.

Plant Model Used (New Plants Italicized)	SPAR(1)SPAR(3) SPAR(2)SPAR(4)	Test-1, Test-2	Test-3, Test-4	Test-5, Test-12	Test-13	Test-14 through Test-40	TSU(1)- TSU(23) BY2-5(1)	Surry-50(1)- Surry-50(5) Com Peak(1)	DEMOM(13) DEMOM(13) BY2-5(4) - Surry 50(6)	DEMOM(10) DEMOM(13) Surry-50(9)	CR3(1) CR3(4)	SLERF, Wolf Creek 302, Peach Bottom 302	SLERF, Heaver Valley	Test-57, Test-62	Test-58, Test-59	Test-60	Test-61	Test-63	Test-65	Test-66
Generate Basic Event Data	X	X	X	X	X	X	X	X	X	X	X									
Generate Change Sets	X	X	X	X	X	X	X	X	X	X	X									
Solve Fault Tree	X	X	X	X	X	X	X	X	X	X	X									
Gather End States by Sequence																				
Gather End States by Cut Set																				
<i>Gather End States by Rules</i>																				
<i>Gather End States by Name</i>																				
Uncertainty - Fault Tree																				
Uncertainty - Sequences - Single																				
Uncertainty - Sequences - Group																				
Uncertainty - End States - Single																				
Uncertainty - End States - Group																				
Importance - Fault Tree																				
Importance - Sequences - Single																				
Importance - Sequences - Group																				
Importance - End States - Single																				
Importance - End States - Group																				
Cut Set Update - Fault Trees																				
Cut Set Update - Sequences																				
Cut Set Update - End States																				
Recover-Fault Trees	X	X	X	X	X	X	X	X	X	X	X									
Recover-Sequences	X	X	X	X	X	X	X	X	X	X	X									
Partition Sequences																				
Link Small / Large Event Trees																				
Logic Save - Fault Tree Editors																				
Version Upgrade	X	X	X	X	X	X	X	X	X	X	X									
Initiating Event Assessment																				
Condition Event Assessment																				
Event Tree Linking																				
Compound Events																				
Base Case Update	X	X	X	X	X	X	X	X	X	X	X									
Calculation Types																				
Uncertainty Distributions																				
N of M Gates																				
Sequence Stress																				
Calculations on Common																				

Plant Model Used (New Plants Italicized)	SPAR(1)SPAR(3) SPAR(2)SPAR(4)	Test-1 Test-2	Test-3 Test-4	Test-5 Test-12	Test-13 Test-40	Test-14 Test-40	Test-42 Test-44 Test-63	SPAR(1) - SURRY(50(5)) Com Peak(1) Com Peak(5)	TESTU(1)- TESTU(23) BY2-5(1) BY2-5(3)	DEMO(1)- DEMO(13) BY2-5(4) - BY2-5(9)	DEMO(10) DEMO(13) SURRY-50(9)	CR3(1) CR3(4)	SLERF, Wolf Creek 302, Peach Bottom 302	SLERF, Beaver Valley	SIMPLE-FT	All 3i Revision Models	TBD	TSTU, Surry-40, Wolf Creek, Peach Bottom, SIMPLE-FT	Wolf Creek, Peach Bottom	TBD	DEMO
Test Reference (New tests italicized) Feature Tested																					
(New features italicized)																					
Cause Plug-ins																					
Results Comparison																					
Verification																					
Load/ Extract Data																					
Fault Tree Paging																					

Idaho National Laboratory		Page: 49 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

3.4 Test Data

The test acceptance criteria ranged from a single value (e.g., total core damage frequency) to hundreds of similar values (e.g., core damage frequency from individual accident sequences) to a set of dissimilar values (e.g., different importance measures for fault trees, moments, and percentiles from uncertainty sampling). In all cases, though, knowledgeable PRA personnel or statisticians at INL obtained and verified the results

3.5 Test Documents

All product documentation is in accordance with Customer procedures documented in the NRC Form 189, Revision 7.

The specific test case macros and their corresponding scenarios are found in Section 13, Appendix A of this document.

The specific test results are found in Section 14, Appendix B of this document. Tests are performed on a plant specific basis, i.e. all tests for the BYRON-2q plant model are performed back to back, and then all the tests for the next plant are performed back to back

3.6 Requirements Validation

Requirements are mapped to the test scenarios through use of a Requirements Traceability Matrix (RTM). See Section 16, Appendix D for the matrix.

The current development phase of the SAPHIRE application is Operations and Maintenance. The entire automated test suite is performed on the application. The test scenarios validate requirements at the system level. A PRA Analyst reviews test data collected from the test scenarios to ensure consistency with predicted results. Test scenarios are added, deleted or revised as additional features are incorporated or the need for updates occurs.

Additionally, prototype testing, unit testing and integration testing will be performed on the areas of code affected by the interface upgrade.

3.7 Test Control Procedures

The various implementations of the SAPHIRE code are distinguished by version and release numbers, displayed in the form “X.Y.” In this notation, the value “X” denotes the version, each of which has one or more major differences that distinguish it from a previous version. The value “Y” denotes the release, which usually is distinguished from its predecessor by one or more minor changes, implemented to correct anomalies or make minor changes in function. The version number can be found within the software by clicking the “About” option from the main Help menu.

The test suit is under configuration control in the SAPHIRE project revision control system repository. All macros, tools and tests scripts that are used to generate automated test results are located in the repository as well. Test results are located in the repository. Upon the initial base line of a test being loaded into the repository, further changes to a particular test are requested through use of the same

Idaho National Laboratory		Page: 50 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

change control logging control database as used for code changes. A more thorough explanation of the SAPHIRE Change Design and Testing Procedure is described in Appendix B of the GEM Interface Revision Project Plan, PLN-1735 Revision 0.

4. Test Phases

Test Phase	Versions Using
Functional	IRRAS 2.5
Comprehensive	IRRAS 2.5
Comprehensive	IRRAS 4.0
Comprehensive	SAPHIRE 5.0 SAPHIRE 6.0
Prototype/Beta	SAPHIRE 6.0 SAPHIRE 7.Y SAPHIRE 8.Y
System test using the hard-coded (e.g., keystroke scripting) automated test suite.	SAPHIRE 6.Y SAPHIRE 7.Y
System test using macro-based automated test suite.	SAPHIRE 7.Y SAPHIRE 8.Y

A thorough Verification and Validation of SAPHIRE 5.0 in the traditional sense was performed in 1995 and documented in NUREG/CR-6116, Volume 9, *Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE) Version 5.0, Verification and Validation (V&V) Manual*, 1995.

Idaho National Laboratory		Page: 51 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

However, this V&V process was found not to be cost-effective because it uses most of the available resources for documentation rather than for actual testing.

Prototype testing of the revised GUI interface involves successful integration with the code, review by a human factors expert, a preliminary release to the customer for evaluation and feedback, then incorporation of any final changes.

System and Acceptance testing is done informally with both in house and with customer experts as a “beta test” for feed back before official version release. Each new SAPHIRE version is beta tested before its release. The beta testers are analysts experienced with PRA methods and terminology, and typically are familiar with earlier versions of SAPHIRE and IRRAS. The primary objective of the beta testing is to verify that the results produced by the new version are correct. The secondary objective is to ensure that the software is user-friendly and functional. Based on feedback from beta testers, programmatic errors and discrepancies are corrected, the user interface has its functionality improved, and new features are recommended. These changes are recorded in the on-line change logging control database at the SAPHIRE Web site.

4.1 Test Design of the Automated Test Suite

The SAPHIRE test uses two different processes. The first process is the development of test scripts or batch files, which are DOS commands that run predetermined macros. The macros are the second process that is used in the testing of the newly released versions of SAPHIRE. Both of these processes along with a simple example will be presented.

The script of batch files are DOS commands that set up the test that will be performed. There are two different types of scripts. The first one shown below is used to execute multiple scripts at once. This script file sets up the output by stating the date and time the test was ran along with what version of SAPHIRE was executed. This script will be considered as the overall test script. The individual lines are DOS commands that help create the output. The first line is an input (i.e., %1 = typed in file name [detail]), which is a file that will be created storing all of the output information. The second line is also a file that contains output information but is a summary report instead of a full detail report (i.e., %2 = typed in file name [summary]). The next group of lines is used to create the headers listed in the detail output report. These lines stamp the output with the date and time of analysis along with the version of SAPHIRE being ran. This information is placed in both the detail and summary report. The last group of lines will now call the individual script files used for specific evaluations. In the case shown the core damage frequency analysis will be performed using the Byron database. The line is a DOS line, which has the core_damage_freq script executed with the output information being stored in %1 (detail file output) and %2 (summary file output) and then which database this script is to be executed on (Byron 2QA). The individual script files will be discussed briefly since they are very similar.

Overall Test Script

```
if %1$==$ goto end
if %2$==$ goto end
```


**Software Acceptance Test Plan
for SAPHIRE Version 8 N6423**

Identifier: INL/EXT-09-16236
Revision: 0
Effective Date: July 21, 2009

```
c:
cd \Saphire7
echo SAPHIRE/GEM Test Suite Summary Report > %2
c:\Saphire7\qatools\datetime "DATE & TIME :" %2
c:\Saphire7\qatools\fversion c:\Saphire7\tools\saphwin.exe %2
echo
>> %2
echo
> %1

rem byron CDF analysis
call scripts\core_damage_freq %1 %2 byrn_2qa
etc.
```

The individual script files can be ran as stand alone or via an overall test script. The individual test scripts are similar except they execute the macros, which tell SAPHIRE what type of analysis is to be performed. The first two lines are the same for the individual test script as for the overall test script. The third line, however, represents the database to be evaluated (%3 = database). The next group of lines is used to create the folder, which the database will be placed and unzipped. Then it executes SAPHIRE and calls the macro, which has the details of the specific analysis. Once the macro has performed its specific analysis the results are dumped into the detail output file (%1) and summary output file (%2). The last line is used to compile all of the outputs together in order to create one large detail file and one large summary file.

Individual Test Script

```
if %1$==$ goto end
if %2$==$ goto end
if %3$==$ goto end

md c:\Saphire7\%3
del c:\Saphire7\%3\*. * /q
c:\Saphire7\qatools\unzip o c:\Saphire7\database\%3.exe d
c:\Saphire7\%3
copy c:\Saphire7\results\%3\qa*.rpt c:\Saphire7\%3
```

Idaho National Laboratory		Page: 53 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

```

call c:\Saphire7\tools\saphwin.exe i386 PROJECT=c:\saphire7\%3\
MACRO=c:\saphire7\macros\Core_Damage_Freq_%3.mac
DETAIL=core_damage_freq_%3.rpt

copy %1 + c:\Saphire7\%3\core_damage_freq_%3.rpt %1

c:\Saphire7\qatools\lastline c:\Saphire7\%3
c:\Saphire7\%3\core_damage_freq_%3.rpt %2

:end

```

The macros are used to perform specific analyses. The following will provide a brief overview of the macros. The macros utilize key words or verbs. The verbs or key words are designed to execute certain functions within SAPHIRE to perform the specific analysis. The following macro that will be discussed is used to solve the fault trees and event tree accident sequences of the specified project database. The macro is core_damage_freq.mac. The macro will be dissected in order for better understanding.

First the macro sets up the analysis. The first line states that no prompt is required prior to starting the analysis (i.e., SAPHIRE will just move down to the execution process instead of waiting for a manual input).

<initial prompt>no</initial prompt>

This part is a comment bracket, which enables the analyst to identify the type of analysis this macro is going to perform and any other pertinent information.

<comment>

TEST CASE NAME: Core Damage Frequency

TEST SCRIPT FILE NAME: Core_Damage_Freq_PWR.mac

GENERAL DESCRIPTION OF WHAT IS VERIFIED:

This test case compares the sequence current case CDF against SAPHIRE version 6 base case results. This test is not plant specific.

NAME OF APPLICATION UNDER TEST: SAPHIRE 7.0

TEST CASE PURPOSE:

Idaho National Laboratory		Page: 54 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

REQUIREMENT(S) VERIFIED: TBD

TEST 01 Solve Fault Trees Fault Tree Probability Results

TEST 02 Solve Sequences Core Damage Frequency Results

TEST CASE ABSTRACT OF TECHNIQUES USED TO TEST THE FEATURE:

The automated tests described herein are grouped to run consecutively.

OTHER FILES REQUIRED TO RUN TEST CASE:

None.

</comment>

The section of the macro sets up the analysis and determines the type of file output along with the name of the file. The %D is a dummy variable, which will read off the name given in the script file that calls this macro.

<setup>

<verbose>

<type>detail</type>

<file name>%D</file name>

</verbose>

</setup>

This part provides a description of the particular scenario that is going to be evaluated. The scenario for this case is %P-01, where %P represents the particular database (i.e., Byrn_2qa) then provides the description of the test (i.e., Solve Fault Trees).

<scenario>

<start>

<name>%P 01</name>

<description>Solve Fault Trees</description>

</start>

</scenario>

Idaho National Laboratory		Page: 55 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

This tells SAPHIRE to go into the Change Set menu and make sure there are no change sets marked then generate the basic event data. Then end this process by using the `</change set>` key word. The “/” tells SAPHIRE to end the process.

`<change set>`

`<unmark></unmark>`

`<generate></generate>`

`</change set>`

The fault tree menu is now executed by using the key word or verb `<fault tree>`. All of the fault trees are marked via the “*” operator, then they are solved at a truncation of 1.0E-16. Once all of the fault trees have been solved a base case update in the random calculation type is performed (key word `<base case update>`, `<analysis>random</analysis>`). The final results are then sent to a file with the name specified (i.e., `ft_current_vs_base.rpt`). This output is then compared to a quality assured set of results to make sure this version of SAPHIRE that is being tested matches the results of a quality assured version of SAPHIRE. Then the fault tree menu option is exited and the scenario is ended.

`<fault tree>`

`<unmark></unmark>`

`<mark mask>*</mark mask>`

`<solve>`

`<truncation>1.0E 16</truncation>`

`</solve>`

`<base case update>`

`<analysis>random</analysis>`

`</base case update>`

`<report>`

`<type>results</type>`

`<sub type>current base</sub type>`

`<file name>ft_current_vs_base.rpt</file name>`

`</report>`

`<compare file>`

`<input 1>ft_current_vs_base.rpt</input 1>`

Idaho National Laboratory		Page: 56 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

```

    <input 2>qa_ft_current_vs_base.rpt</input 2>
    <output>compare.rpt</output>
</compare file>

```

```

<report>
  <type>results</type>
  <sub type>current only</sub type>
  <file name>ft_current_only.rpt</file name>
</report>

```

```

<compare file>
  <input 1>ft_current_only.rpt</input 1>
  <input 2>qa_ft_current_only.rpt</input 2>
  <output>compare.rpt</output>
</compare file>
</fault tree>
<scenario><end></end></scenario>

```

This next part provides a description of the particular scenario that is going to be evaluated. The scenario for this case is %P-02, where %P represents the particular database (i.e., Byrn_2qa) then provides the description of the test (i.e., core damage frequency test).

```

<scenario>
  <start>
    <name>%P 02</name>
    <description>Core Damage Frequency Test</description>
  </start>
</scenario>

```

Idaho National Laboratory		Page: 57 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

This tells SAPHIRE to go into the Change Set menu and make sure there are no change sets marked then generate the basic event data. Then end this process by using the </change set> key word.

<change set>

<unmark></unmark>

<generate></generate>

</change set>

The sequence menu is now executed by using the key word or verb <sequence>. All of the sequences are marked via the “*” operator, then they are solved at a truncation of 1.0E-16. Once all of the sequences have been solved a base case update in the random calculation type is performed (key word <base case update>, <analysis>random</analysis>). The final results are then sent to a file with the name specified (i.e., sq_current_vs_base.rpt). This output is then compared to a quality assured set of results to make sure this version of SAPHIRE that is being tested matches the results of a quality assured version of SAPHIRE. Then the sequence menu option is exited and the scenario is ended.

<sequence>

<unmark></unmark>

<include>

<mark event tree mask>*</mark event tree mask>

<mask operation>and</mask operation>

<mark sequence mask>*</mark sequence mask>

<mask operation>and</mask operation>

<mark logic fault tree>*</mark logic fault tree>

</include>

<solve>

<truncation>1.0E 16</truncation>

</solve>

<base case update>

<analysis>random</analysis>

Idaho National Laboratory		Page: 58 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

</base case update>

<report>

<type>results</type>

<sub type>current base</sub type>

<file name>sq_current_vs_base.rpt</file name>

</report>

<compare file>

<input 1>sq_current_vs_base.rpt</input 1>

<input 2>qa_sq_current_vs_base.rpt</input 2>

<output>compare.rpt</output>

</compare file>

<report>

<type>results</type>

<sub type>current only</sub type>

<file name>sq_current_only.rpt</file name>

</report>

<compare file>

<input 1>sq_current_only.rpt</input 1>

<input 2>qa_sq_current_only.rpt</input 2>

<output>compare.rpt</output>

</compare file>

</sequence>

<scenario><end></end></scenario>

The last line of every macro is the exit program verb. This command exits from SAPHIRE in order for another macro can be executed.

Idaho National Laboratory		Page: 59 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

<program exit></program exit>

The above provided only a brief description of how the verbs or key words work in the macros. However, all of the other menus and actions that can be performed by SAPHIRE (i.e., End state evaluations, importance measures, GEM evaluations, etc) can be created using the same format. State the starting key word or verb (<end state>), then add the type of evaluation required (<solve> [i.e., gather cut sets]), then end the process by adding “/” to the verb (</end state>).

By developing the key words or verbs directly into SAPHIRE allows for a quick and direct new version verification.

The automated test suite utilizes embedded software hooks in the API to utilize the application code to run the test macros. The test macros were developed to replace the previous test tool that is not longer being supported by the vendor. The original test sequences were translated into the macro language and then rerun on the SAPHIRE 7.Y software release to ensure results matched the pre macro results. The pre and post macro results were independently verified and validated by a PRA analyst.

For each test result, the first line of the test result identifies the test ID and description along with the time at which the particular test was started. This is illustrated below in the sample test result (e.g., SURRY-50-05). Then, after the identifier line, the steps processed by the test are shown. In the example, the SURRY-50 sequences are solved using a truncation of 1E-9/yr and then recovery rules are applied. The cut sets are run through a cut-set update. Then, the test gathers end-state cut sets via the partition rules (again with 1E-9/yr truncation). These end-state cut sets are updated. Lastly, the results are compared against the stored “correct” results for the end states of AD5, AD6, AH1, and S2D1. If the results match the “correct” results, a “pass” is indicated, otherwise a “failed” would be indicated. Then, the time of test completion is recorded.

SURRY–50-05 Scenario: Check End State Cut Sets started at 12:48:28 AM

Sequences solved

with prob cut off (1.0E-09) and with recovery

Sequence cut sets updated

End States gathered by cut set partition

with prob cut off (1.0E-09)

End State cut sets updated

END STATE CUTSET RESULTS:

AD5 pass

AD6 pass

AH1 pass

Idaho National Laboratory		Page: 60 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

S2D1 pass

Scenario: Check End State Cut Sets completed at 12:50:05 AM.

In addition to specific test scenario data, the user ID of the person running the test; the version of SAPHIRE being tested; and the version of the operating system are automatically recorded.

Final acceptance of any documentation, code, and test results is considered complete when all parties sign off on the completed change.

4.2 Sources of Data

There are 66 different tests (which increase as new tests are designed) that have been identified and defined as part of the automated test suite. For each test, criteria are developed to determine if SAPHIRE accomplished a task. This generation of acceptance criteria results in a significant amount of information, since a test may use multiple PRA models. For example, the first test (Test-01) is performed using 82 different databases. Also, where applicable, the test evaluated the different mechanisms in SAPHIRE to accomplish the same task. An example of this aspect is the ability to generate end state cut sets using either the predefined end state categories (on the event tree) or using the end state partition rules.

4.3 Entrance and Exit Criteria

The entrance criteria for testing are to obtain the stored test repository and associated project databases from the revision control system. The test scripts will be exercised as noted in the script.

The exit criteria for testing are to check the suite test output results file for any failed tests. Note that one test has been designed to always fail and is used as a “false positive” results to ensure functionality of the designed comparison against the QA benchmarked results.

4.4 Requirements

The requirements validated are listed at the beginning of each test macro script.

4.5 Work Products

See NRC Form 189, Revision 7, for a list of the software work product deliverables for this effort.

Idaho National Laboratory		Page: 61 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

5. Development and Test Environment

5.1 Hardware Platform

Both development and test were performed on the same configuration of hardware.

The testing was performed on a stand alone PC with the following hardware requirements:

A Dell Optiplex GX270 with

- Windows XP Professional, with current service packs installed.
- An Intel Pentium(R) 4 CPU 2.60 GHZ processor.
- 512 MB of RAM
- Super VGA (800 x 600) video adapter and monitor.
- Standard Keyboard, and mouse.

A network connection was used to transfer files necessary to perform the tests.

5.2 Software Platform

The software packages required for the development and test environments are:

- Borland DELPHI Version 2006, IDE and run time engine.
- The following third party add-ons were used in the development of the tool set:

3rd Party Add-on	Description
Abbrevia305Beta6	Abbrevia is a compression toolkit for Borland Delphi, C++Builder, & Kylix.
Compilers2006.inc	Include file to determine which compiler is currently being used to build the project/component
DIRegExInstall	DIRegEx is a library of Delphi components and procedures that implement regular expression pattern matching.
Gif imaged7c	TGIFImage ported to Delphi, supports animated GIF files in Delphi.

Idaho National Laboratory		Page: 62 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

3rd Party Add-on	Description
JVCL300Help-HTML	Jedi Visual Component Library
JVCL320CompleteJCL197	Jedi Visual Component Library JVCL is a library of more than 600 visual and non-visual components for Delphi 5, 6, 7, 2005 and 2006; C++Builder 5, 6 and 2006 (Borland Developer Studio).
mxCaptionButton	?????
PDF-viewer for Delphi	A VCL component capable of printing PDF files and displaying PDF files on Delphi forms
pngimage	An Image widget that overcomes PNG browser incompatibilities.
QPDFFrontEnd	?????
quickpdffull	Creates PDF documents within applications written on Delphi or C++ Builder.
smcmpnt	Delphi freeware SMComponent library. Send Mail component library.
SynEdit	A syntax highlighting editor for programming languages
TeeChart	A a charting and plotting utility.
VirtualTreeView	Virtual Treeview is a treeview control.

- Stonybrook Modula-2 Version 4, IDE and run time engine.
- SAPHIRE Version 8 release.
- SAPHIRE Version 8 automated test suite, scripts and SQA tools.

5.3 Test Location

In-house testing for unit, integration, system, acceptance, and prototype testing occurs at the INL EROB facility. SAPHIRE has a test lab located in the EROB building, in cubical 2EM403.

Idaho National Laboratory		Page: 63 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

Out of house testing will be performed by NRC PM selected users and locations for the prototype application. The selected users are TBD at this time.

Out of house testing will be performed by NASA selected users and locations for the prototype application. The selected users are TBD at this time.

5.4 Staffing and Training

Test performers and test developers of the team need to be familiar with DELPHI © Version 7 IDE, Modula-2 IDE, DOS batch files and scripting, and use of the Microsoft Windows platforms to support the interface upgrade and the functional improvement development and testing effort.

5.5 Test Summary

The results of the tests, including their output are presented in Appendix B.

6. Schedule

See NRC Form 189, Revision 7, for Job Code Y6394, Maintain and Support SAPHIRE Code and Library of PRA document for detailed schedule activities.

Idaho National Laboratory		Page: 64 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

A.

A. Appendix A Automated Test Scripts

Idaho National Laboratory		Page: 65 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

Idaho National Laboratory		Page: 66 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

Appendix A Automated Test Scripts

This appendix will contain information on the macro tests scripts run on SAPHIRE Version 8. Here is a sample test script for the draft of the document. Copies of the test scripts are stored in the revision control system for review purposes.

A-1. Events and Condition Assessment Tests for Arkansas Nuclear 01(AN01)

<initial prompt>no</initial prompt>

<comment>

TEST CASE NAME: Events and Condition Assessment Tests

TEST SCRIPT FILE NAME: Cond_Asses_ANO1_301.mac

GENERAL DESCRIPTION OF WHAT IS VERIFIED:

Perform GEM Events and Condition Assessment against SAPHIRE/GEM Version 7.0 results.

NAME OF APPLICATION UNDER TEST: SAPHIRE 8.0

TEST CASE PURPOSE:

REQUIREMENT (S) VERIFIED: TBD

TEST-03 EFW out of service for 72 hours.

TEST-04 Emergency Diesel Generator out of Service for 3 months.

TEST CASE ABSTRACT OF TECHNIQUES USED TO TEST THE FEATURE:

The automated tests described herein are grouped to run consecutively.

OTHER FILES REQUIRED TO RUN TEST CASE:

cond_asses.bat batch file

qa_tools: DateTime.exe, FVersion.exe, LastLine.exe, Unzip.exe

</comment>

<setup>

<verbose>

<type>detail</type>

Idaho National Laboratory		Page: 67 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

```

    <file name>%D</file name>
  </verbose>
</setup>

<scenario>
  <start>
    <name>%P-03</name>
    <description>Condition EFW out of service for 72 hours</description>
  </start>
</scenario>

<condition>
  <unmark></unmark>
  <delete>
    <name>EFW-72HRS</name>
  </delete>

  <add>
    <name>EFW-72HRS</name>
    <description>EFW out of service for 72 hours</description>
    <single>
      <event name>EFW-PMP-CF-RUN</event name>
      <calc type>T</calc type>
    </single>
  </add>

  <mark name>EFW-72HRS</mark name>
  <process>
    <duration>72</duration>
    <truncation>1.0E-15</truncation>

    <report>
      <type>results</type>
      <sub type>current only</sub type>
      <file name>efw_72hrs.rpt</file name>
    </report>
  </process>

  <compare file>
    <input 1>efw_72hrs.rpt</input 1>
    <input 2>qa_efw_72hrs.rpt</input 2>
    <output>compare.rpt</output>
  </compare file>
</condition>

```


Idaho National Laboratory		Page: 68 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

<scenario><end></end></scenario>

<scenario>
<start>
<name>%P-04</name>
<description>Condition EDG out of service for 2190 hours or 3 months</description>
</start>
</scenario>

<condition>
<unmark></unmark>
<delete>
<name>EDG-2190HRS</name>
</delete>

<add>
<name>EDG-2190HRS</name>
<description>EDG out of service for 2190 hours or 3 months</description>
<single>
<event name>EPS-DGN-FS-DG1</event name>
<calc type>T</calc type>
</single>
</add>

<mark name>EDG-2190HRS</mark name>
<process>
<duration>2190</duration>
<truncation>1.0E-15</truncation>

<report>
<type>results</type>
<sub type>current only</sub type>
<file name>edg_3months.rpt</file name>
</report>
</process>

<compare file>
<input 1>edg_3months.rpt</input 1>
<input 2>qa_edg_3months.rpt</input 2>
<output>compare.rpt</output>
</compare file>
</condition>

<scenario><end></end></scenario>

Idaho National Laboratory		Page: 69 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

<program exit></program exit>

Idaho National Laboratory		Page: 70 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

Idaho National Laboratory		Page: 71 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423	Identifier:	INL/EXT-09-16236
	Revision:	0
	Effective Date:	July 21, 2009

B. Appendix B Automated Test Results

**Software Acceptance Test Plan
for SAPHIRE Version 8 N6423**

Identifier: INL/EXT-09-16236
Revision: 0
Effective Date: July 21, 2009

Appendix B

Automated Test Results

This section will contain an example of the formal test results obtained from the automated test suite described in the previous section. A complete set of test results for each incremental version are stored for historical review.

B-1. Class Change Tests for DEMO Plant results

DEMO-11

SEQUENCE CUT SETS (QUANTIFICATION) REPORT

Project: DEMO

Case : Current

Analysis: RANDOM

Event Tree	Sequence	% Total	Cut Set %	Prob./Freq. Inputs
LOSP 2	95.0 95.0	2.0E-002	DG-A	
	99.8 4.8	1.0E-003	E-MOV-1	
	99.9 0.1	2.5E-005	E-MOV-A, E-MOV-B	
	100.0 0.1	1.5E-005	E-MOV-A, E-PUMP-B	
	100.0 0.1	1.5E-005	E-MOV-B, E-PUMP-A	
	100.0 0.0	9.0E-006	E-PUMP-A, E-PUMP-B	
	100.0 0.0	5.0E-007	E-CV-A, E-MOV-B	
	100.0 0.0	5.0E-007	E-CV-B, E-MOV-A	
	100.0 0.0	3.0E-007	E-CV-A, E-PUMP-B	
	100.0 0.0	3.0E-007	E-CV-B, E-PUMP-A	
	100.0 0.0	1.0E-008	E-CV-A, E-CV-B	

2004/08/17 Page # 14:01:39

Model Rev. /- /--

DEMO-12

SEQUENCE CUT SETS (QUANTIFICATION) REPORT

Project: DEMO

Case : Current

Analysis: RANDOM

Event Tree	Sequence	% Total	Cut Set %	Prob./Freq. Inputs
LOSP 2	99.1 99.1	1.0E-001	DG-A	
	100.0 1.0	1.0E-003	E-MOV-1	
	100.0 0.0	2.5E-005	E-MOV-A, E-MOV-B	
	100.0 0.0	1.5E-005	E-MOV-A, E-PUMP-B	
	100.0 0.0	1.5E-005	E-MOV-B, E-PUMP-A	
	100.0 0.0	9.0E-006	E-PUMP-A, E-PUMP-B	
	100.0 0.0	5.0E-007	E-CV-A, E-MOV-B	
	100.0 0.0	5.0E-007	E-CV-B, E-MOV-A	
	100.0 0.0	3.0E-007	E-CV-A, E-PUMP-B	
	100.0 0.0	3.0E-007	E-CV-B, E-PUMP-A	
	100.0 0.0	1.0E-008	E-CV-A, E-CV-B	

2004/08/17 Page # 14:01:39

Model Rev. /- /--

Idaho National Laboratory		Page: 73 of 77
Software Acceptance Test Plan for SAPHIRE Version 8 N6423		Identifier: INL/EXT-09-16236 Revision: 0 Effective Date: July 21, 2009

DEMO-13

SEQUENCE CUT SETS (QUANTIFICATION) REPORT

Project: DEMO

Case : Current

Analysis: RANDOM

Event Tree	Sequence	% Total	Cut Set %	Prob./Freq. Inputs
LOSP 2	38.5 38.5	1.0E-001	DG-A	
	76.9 38.5	1.0E-001	E-MOV-1	
	80.8 3.9	1.0E-002	E-CV-A, E-CV-B	
	84.6 3.9	1.0E-002	E-CV-A, E-MOV-B	
	88.4 3.9	1.0E-002	E-CV-A, E-PUMP-B	
	92.3 3.9	1.0E-002	E-CV-B, E-MOV-A	
	96.1 3.9	1.0E-002	E-CV-B, E-PUMP-A	
	100.0 3.9	1.0E-002	E-MOV-A, E-MOV-B	
	100.0 3.9	1.0E-002	E-MOV-A, E-PUMP-B	
	100.0 3.9	1.0E-002	E-MOV-B, E-PUMP-A	
	100.0 3.9	1.0E-002	E-PUMP-A, E-PUMP-B	

2004/08/17 Page # 14:01:39
Model Rev. /- /--

DEMO-14

SEQUENCE CUT SETS (QUANTIFICATION) REPORT

Project: DEMO

Case : Current

Analysis: RANDOM

Event Tree	Sequence	% Total	Cut Set %	Prob./Freq. Inputs
LOSP 2	95.0 95.0	2.0E-002	DG-A	
	99.8 4.8	1.0E-003	E-MOV-1	
	99.9 0.1	2.5E-005	E-MOV-A, E-MOV-B	
	100.0 0.1	1.5E-005	E-MOV-A, E-PUMP-B	
	100.0 0.1	1.5E-005	E-MOV-B, E-PUMP-A	
	100.0 0.0	9.0E-006	E-PUMP-A, E-PUMP-B	
	100.0 0.0	5.0E-007	E-CV-A, E-MOV-B	
	100.0 0.0	5.0E-007	E-CV-B, E-MOV-A	
	100.0 0.0	3.0E-007	E-CV-A, E-PUMP-B	
	100.0 0.0	3.0E-007	E-CV-B, E-PUMP-A	
	100.0 0.0	1.0E-008	E-CV-A, E-CV-B	

2004/08/17 Page # 14:01:39
Model Rev. /- /--

