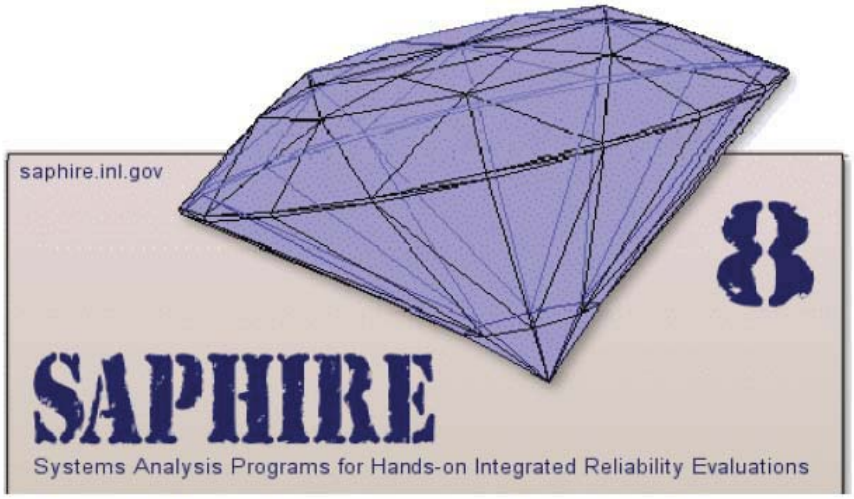
INL/EXT-09-16696
Rev. 1

SAPHIRE 8 Software Configuration Management Plan

January 2010



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

Idaho National Laboratory

INL/EXT-09-16696
Rev. 1

SAPHIRE 8 SOFTWARE CONFIGURATION MANAGEMENT PLAN

January 2010

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Nuclear Regulatory Commission
Washington, DC 20555
Project No. N6423

REVISION LOG

Revision Number	Effective Date	Affected Pages	Description of Change
0		0	Initial issue (Preliminary Configuration Management Plan)
1			Updated headers and text in Section 1.3 to match project plan information

SIGNATURES

Approved by

Curtis Smith	1/26/2010
Curtis L. Smith, INL Project Manager	Date
S. Ted Wood, SAPHIRE 8 Support Manager	Date

CONTENTS

1. Introduction

1.1 Project Background and Objectives

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 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.

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 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, "Quality Assurance Requirements", 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-13620, 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:

- 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 uses the document Software Quality Assurance Plan Document ID: INL/EXT-09-16697 for SQA.

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 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 became 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 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, monthly reports, 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.

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.

1.3 Configuration Management Approach

The INL software developers use version control for both the formally released SAPHIRE versions, as well as for source code. For each formal release of the software, the developers perform an acceptance test: the software must pass a suite of automated tests prior to official release. IV&V tests and test observations are also being performed as given in the IV&V plan to support the first general release version of SAPHIRE 8.

Each official release of SAPHIRE is assigned a unique version identifier. The release is bundled into a standard installation package for easy and consistent set-up by individual users. Included in the release is a list of bug fixes and new features for the current release, as well as a history of those items for past releases. Each formal release of SAPHIRE will have passed an acceptance test described in the Software Acceptance Test Plan (INL/EXT-09-16236).

In addition to assignment of a unique version identifier for an official software release, each source code file is kept in a controlled library (a revision control system, or RCS). (Source code is a collection of all the computer instructions written by developers to create the finished product.) The library is kept on a server, where back-ups are regularly made. (Individual developers/programmers machines are periodically backed up as well.)

The source code version control library requires that individual programmers "check-out" all files that they intend to modify. Prior to "check-in", programmers must document changes. A record is kept of all changes, both as explained by the developer, and as individual copies of each version of a file. At any time, the developer can retrieve past versions intact, if necessary.

The SAPHIRE software program is continually modified, in response to user reported bugs and suggestions, and contractually specified enhancements. The version control procedure described above ensures a methodical approach to tracking and releasing these changes.

Quality assurance reviews configuration management and control processes to ensure that only authorized changes are made to the software. All software modules that have been tested, documented, and approved for inclusion into the next release of the software are baselined. The software/system database "librarian" controls the baselined source code. Copies of current build routines needed to construct the software, including all copies of all build routines used in all prior releases are also under the librarian control.

SAPHIRE uses a configuration management database as a control library for all information related to the development of software fixes, enhances, baselines, and subsequent releases. Processes are in place to uniquely identify all components, modules, documentation, error reports, test suites, and test results through the establishment of a configuration control tracking number.

Bug fixes and all supporting documentation are placed under configuration control. Notes from the reporting user are obtained describing the general context of the bug, as well as step-by-step actions to reproduce the bugs. This includes acquiring a copy of the user's database, when necessary. The bug is classified and prioritized according to severity. A bug is considered "minor" if it inconveniences the user, but a workaround exists to produce a correct answer. A bug is "major" if it prevents the user from obtaining the correct answer. Bugs found in more commonly used features are considered a higher priority than those found in less used features. User deadlines are also considered. Bug fixes are tested in the environment in which they were reported, as well as other places if possible side effects are suspected. Sometimes, a release candidate is made available to the reporting user or group of users to ensure that the problem has been satisfactorily fixed. Once a bug has been resolved, it is added to the list of changes for the next official version, which must pass the set of acceptance tests described in the next section.

Software enhancements and supporting requirements and documentation are also placed under configuration control. Enhancements are prioritized and implemented, with intermediate testing by the developer and often by the requestor. Once the process and results appear acceptable, the feature is added to the next official release.

1.4 Release Management

Upon the need to release an updated version of SAPHIRE, the release management process described in Figure 1 is used to control the process. As part of the testing, a testing witness form is used to log the results of the testing process; this form is shown in Figure 2.

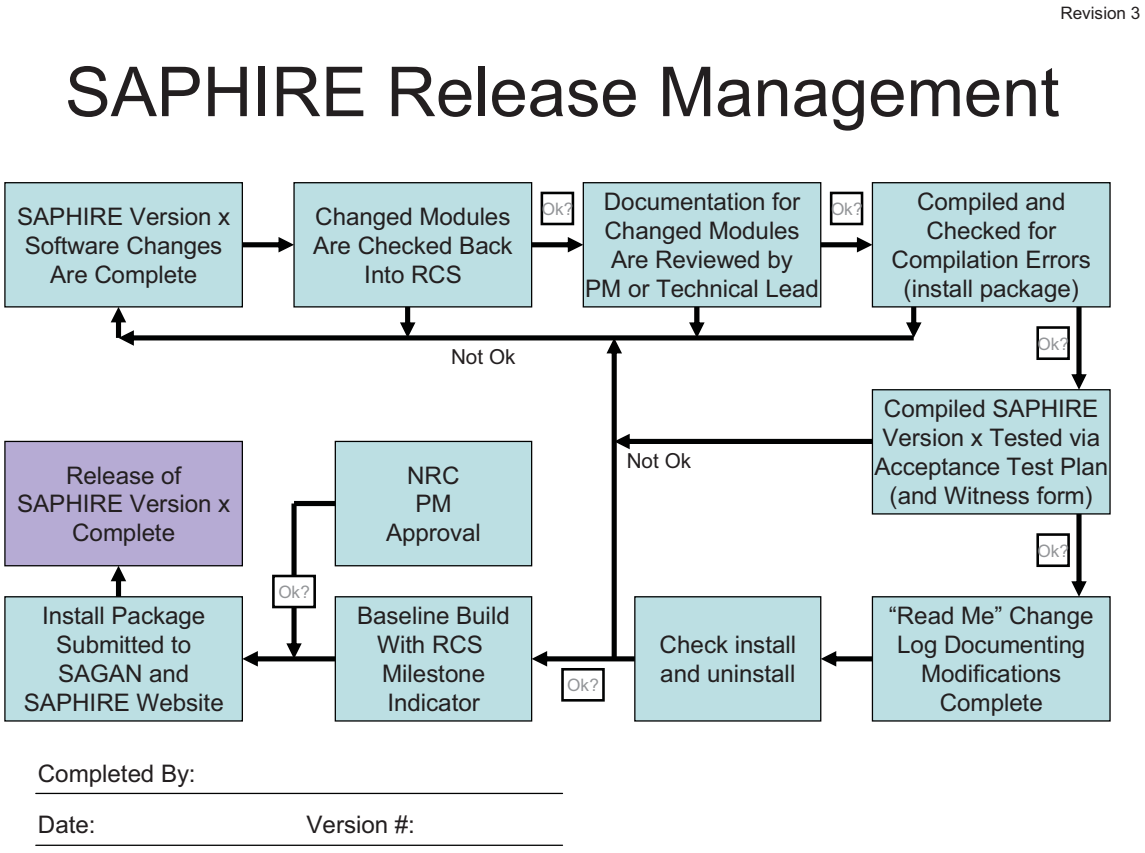


Figure 1. SAPHIRE release management process.

Revision 2

SAPHIRE Test Witness Monitor Form

Name of Witness:	
Signature of Witness:	
Date:	
Time:	
Software Version being tested:	
(Name and Build Number Version)	
Software Test Procedure(s) used:	
(Name and Version)	
Test Platform(s) Used:	
Test type:	Automated [ ] Manual [ ]
Test Results:	Pass [ ] Fail [ ]

PRE-TEST ACTIVITIES	**STATUS**
Test Schedule Established	YES [ ] No [ ] N/A [ ]
Test Procedures Reviewed	YES [ ] No [ ] N/A [ ]
Test Environment setup in accordance with Test Procedures	YES [ ] No [ ] N/A [ ]
Test Activities	
Test Procedures are used	YES [ ] No [ ] N/A [ ]
Test Environment used	YES [ ] No [ ] N/A [ ]
Test Deviations documented	YES [ ] No [ ] N/A [ ]
Test Nonconformance documented	YES [ ] No [ ] N/A [ ]
Post Test Activities	
Verify Test Deviations Documented	YES [ ] No [ ] N/A [ ]
Test Deviations Corrected	YES [ ] No [ ] N/A [ ]
Additional Comments	

Figure 2. SAPHIRE test witness monitor form.