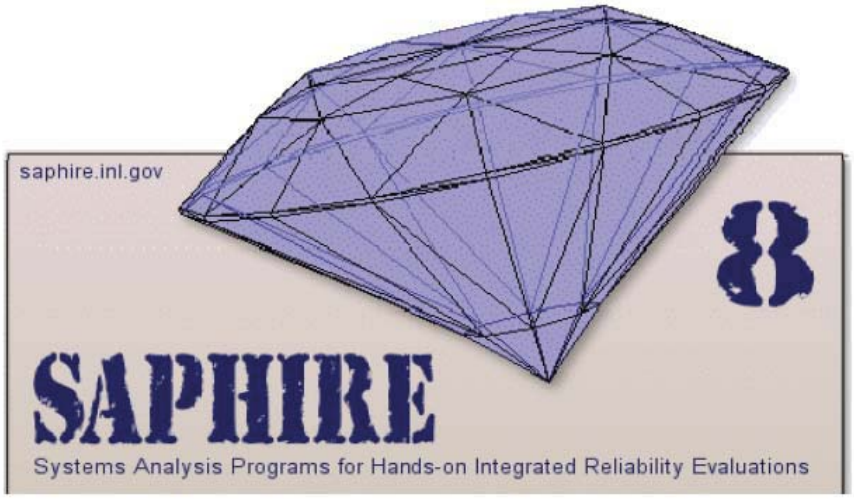
INL/EXT-09-16697
Rev. 1

SAPHIRE 8 Software Quality Assurance Plan

February 2010



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

Idaho National Laboratory

INL/EXT-09-16697
Rev. 1

SAPHIRE 8 Software Quality Assurance Plan

February 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

SOFTWARE Quality Assurance Plan for SAPHIRE Version 8 N6423	Identifier: INL/EXT-09-16697 Revision: 1 Effective Date: 2/17/10

REVISION LOG

Revision Number	Effective Date	Affected Pages	Description of Change
0		0	Initial issue (Preliminary Quality Assurance Plan)
1	2/17/2010	Multiple	Editorial changes based upon IV&V review.

SIGNATURES

Approved by

Curtis Smith	2/17/10
Curtis L. Smith, INL Project Manager	Date
S. Ted Wood, SAPHIRE 8 Support Manager	Date

CONTENTS

1. Introduction

1.1 Project Background and Objectives

This Quality Assurance (QA) Plan documents the QA activities that will be managed by the Idaho National Laboratory (INL) related to JCN N6423.

The Nuclear Regulatory Commission (NRC) developed the SAPHIRE computer code for performing probabilistic risk assessments (PRAs) using a personal computer (PC) at the 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 Standardized Plant Analysis Risk (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.

1.2 Project Scope and Organization

All NRC work assigned to a Department of Energy (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 PDD-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:

- 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. SAPHIRE 8 is using this plan, Software Quality Assurance Plan Document ID: INL/EXT-09-16697.

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, 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 Office of Nuclear Regulatory Research (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

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 Project Manager (PM) and Technical Monitor (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 once a year against the requirement of NUREG/BR-0167.

1.3 Quality Assurance and Assessment Approach

Each JCN will have a quality level determination made by a Quality Level Analyst in accordance with LWP-13014 and documented on INL Form 414.A89. The quality level for SAPHIRE 8 has been determined to be Quality Level 3 (Low). The Quality Level Determination (QLD) has been entered into the INL Enterprise Management System as part of the Quality Level Database. The QLD database entry for SAPHIRE 8 is stored as ALL-000594.

Product quality is a key component of SAPHIRE. The SAPHIRE QA processes documented in the report provides the basis for setting quality objectives, progress, and the necessary framework for quality improvements. This QA plan will evolve as the SAPHIRE product is enhanced to provide the end user with solutions to their technical problems and cost-effectively meet user expectations. A majority of the changes within the SAPHIRE software occur because

the end user has identified characteristics that provide "new potential", thus resulting in SAPHIRE evolving as each new feature is discovered and implemented. Therefore, the majority of software maintenance comes about not because of deficiencies in the code, but because it was modified to embrace improved methods for risk and reliability assessment.

In order to ensure the quality of the SAPHIRE software, the INL uses a variety of software development methods, including:

- Controlling software versions for both the formally released SAPHIRE versions, as well as for source code.
- Following a standard approach to bug fixes and new features.
- Using a cyclical design process to prototype changes.
- Performing acceptance tests that the software must pass prior to official release.

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 any changes made. A record is kept of all changes, both as documented 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. Since the SAPHIRE software program is continually modified, the version control procedure ensures a methodical approach to tracking and releasing these changes.

As new features and bug fixes are made, the INL developers follow a standard approach to integrating these items into SAPHIRE. For bug reports (via the anomaly tracking system), the developers take notes from the user describing the general context of the bug, as well as step-by-step actions to reproduce the bugs. This bug information includes acquiring a copy of the user's database, when necessary. Then, 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. Software enhancements follow much the same approach as bug fixes. 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.

The level of effort for the software design process corresponds to the size and complexity of the proposed change. Developers use a cyclical prototyping design methodology as a means to clarify and refine the change. The prototyping process involves the requestor throughout development. The developers will interact with the requestor(s) both initially and throughout the design and development process to ensure the change accomplishes the expected goal.

Prior to any official SAPHIRE release of versions 8, the software is run through the tests in the Acceptance Test Plan (INL/EXT-09-16236). The tests simulate user input to the computer

through a test script, and results are captured and compared to expected results. This ensures that given a static input PRA file, the risk or reliability results from SAPHIRE will be consistent from one release to the next. These acceptance tests were developed by first identifying the critical tasks performed in a PRA. Then these tasks were mapped to the SAPHIRE functions that perform these tasks. The critical functions were determined to include the following:

1. Fault tree analysis
2. Event tree and sequence analysis
3. End state analysis
4. Importance measures analysis
5. Uncertainty analysis
6. Change sets
7. Data utility functions
8. Workspace module functionality

The SAPHIRE QA process encompasses several activities the INL uses to ensure quality throughout the development cycle. The elements of these activities are illustrated in Figure 1.

Figure 1. SAPHIRE quality assurance elements.

As part of the overall QA process, the SAPHIRE Testing, Verifying, and Validating (TV&V) process and results were previously documented in NUREG/CR-6688, "Testing, Verifying, and Validating SAPHIRE Versions 6.0 and 7.0 (Smith et al, 2000). Within that document, Section 1 explains that the version 6 and 7 TV&V departs from earlier Verifying and Validating (V&V) efforts (for versions 4 and 5) by focusing on the development and execution of a set of automated test scripts. This TV&V process was expanded and automated so that the validity of the core functionality of SAPHIRE can be verified on an ongoing basis with each incremental release. The QA process for Version 8 builds upon this TV&V approach and is planned to be documented in a future SAPHIRE NUREG/CR update.

1.4 Software Releases

A *released* version of SAPHIRE represents an incremental version of the "current release" that is made generally available. Note that at times, significant enhancements and additions were introduced as part of these released versions, so while existing bugs may be fixed, it is possible that new bugs are introduced via these new features. Nonetheless, for each incremental version, the SAPHIRE software must pass an extensive automated test process to ensure that existing calculation features are not compromised. Definitions of the software release terms used by the SAPHIRE development team include:

Beta	The "beta" version of SAPHIRE is that numbered version (e.g., 8.x) that is currently under development at the INL. This version is used to add new features and to make significant modifications to either the analysis or user interface portions of the software. Since this version is in development, it is possible that features are incomplete or modification may leave the software in an unstable state. In addition, the software documentation may not be available specific to this version of the software. This version is not available for general release.
Current Release	The "current release" version of SAPHIRE is the most recent numbered version of the software that is "frozen." The term "frozen" indicates that the analysis and user interface portions of the software will not be modified, with the exception of needed changes related to programming errors or limitations. Typically, the current release is the version that undergoes the largest amount of use, and consequently, has the highest degree of testing.
N-1 Release	The "N-1 release" version of SAPHIRE is the second-to-last released "frozen" version.

Note that for all versions of SAPHIRE, transfer of the software or related information (in electronic or hardcopy format) is prohibited unless prior approval is obtained since the software is subject to U.S. export control regulations.

1.5 Change Design and Testing Procedure

Software developers follow the SAPHIRE Change Design and Testing Procedure (INL/EXT-10-17915) when adding a new feature or revising an existing capability. This procedure first describes the general approach to changes, and then describes processes that are more specific. The process stages include design and development, testing, and documentation. The initial design effort corresponds to the size and complexity of the change. Developers use a cyclical prototyping design methodology as a means to clarify and refine the change. The prototyping process involves the requestor throughout development. The developers will interact with the

requestor(s) both initially and throughout the design and development process to ensure the change accomplishes the expected goal.

Changes and additions to the software vary from very small bug fixes to significant enhancements and new capabilities. The complexity of a change or addition also varies by item. Therefore, the developers use a graded approach to design. They spend more time and effort on larger and/or more complex changes than on relatively simple items. Areas of changes or bugs also dictate the level of effort. For example, problems in cut set generating are much more important than problems in report areas. Enhancements to cut set generation are researched much more carefully than enhancements to reports.

The frequency and formality of communications with the requestor also corresponds to the size and complexity of the change. This ensures that time and money is spent wisely.

The first step in designing a change to SAPHIRE requires that the developers and requestors define and discuss the problem and propose a solution. The developer should gain a broad understanding of the goal of the change, and the requestor should understand in general terms how the proposed solution will accomplish the goal.

At this point in the process, the change will be summarized in a SAPHIRE Change Design Form (see Figure 2), where the problem will be summarized and categorized.

Once a clear definition of the change has been identified, additional items are considered, including:

- When applicable, define the necessary inputs and expected outputs.
- Determine the approximate complexity and level of effort required to accomplish the task.
- Consider how existing code functionality can be leveraged to help accomplish the task.
- Consider potential effects on other parts of SAPHIRE.

The next step is to prove the concept. This means developing key internal functions as well as a rudimentary interface to access and test those functions. This step serves to test the feasibility of the solution, and helps the designers understand the problem. The results of this step are used for further discussion between the developer and the requestor. This is considered the first iteration of the prototype. Depending upon the results, the design may be modified and refined. The prototype will be modified or rewritten to reflect the information learned.

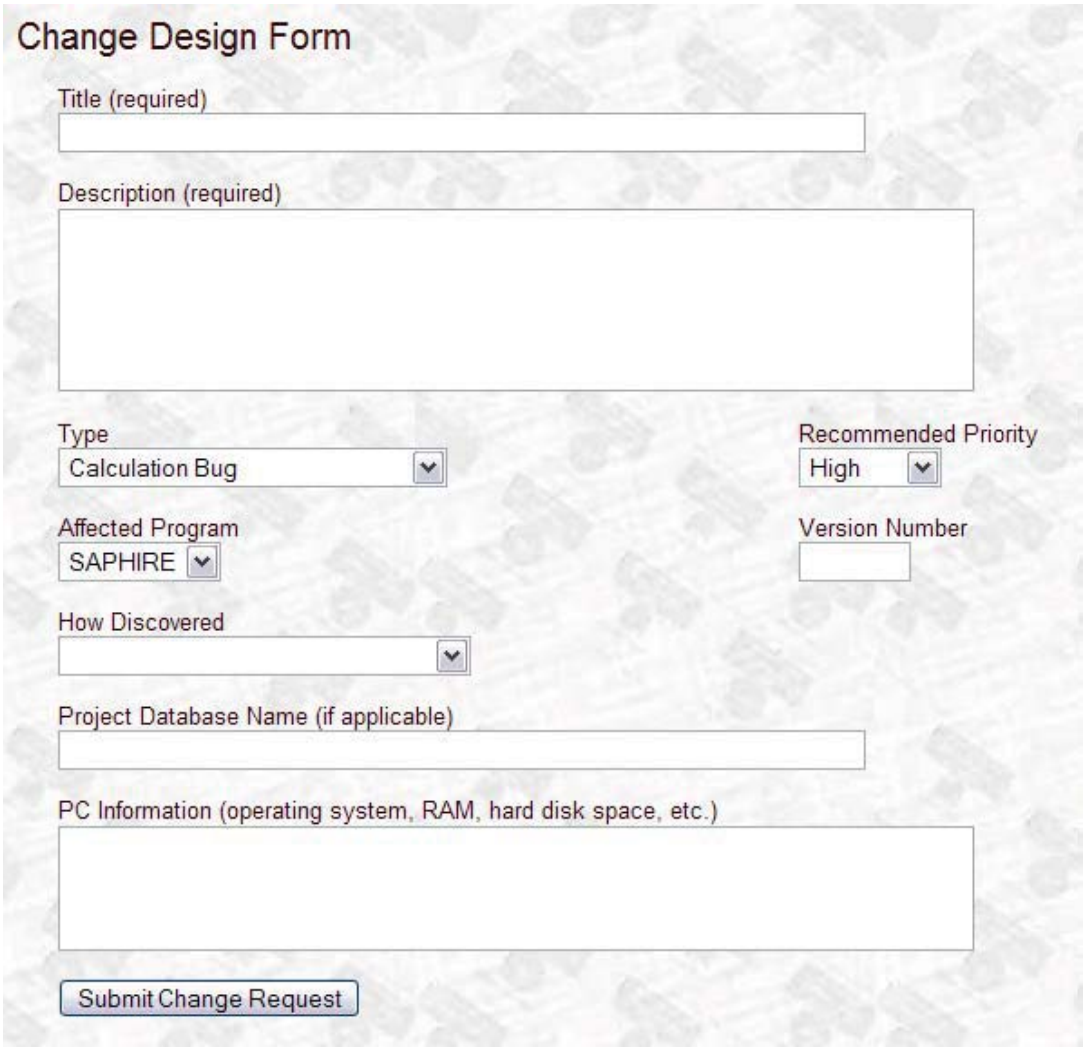


Figure 2. SAPHIRE 8 Change Design Form.

An iteration of the software should improve the functionality of the change to bring it closer to its goal. Successive passes, as the design and prototype stabilize, will incorporate more and more of the following items:

- Additional supporting functions
- Refined and more complete user interface
- Integration into the SAPHIRE user interface
- Auxiliary functions to facilitate ease of use

Auxiliary functions are niceties that contribute to ease of use. They vary according to the task, but may generally include such things as customizing, sorting, and/or saving data, generating reports, loading and extracting data between projects, toolbar short-cuts, and individual and bulk processing of data. These types of auxiliary functions are added as time and budget permit. Depending on the scope and complexity of the task, the requestor and the developer maintain contact throughout the development process. Specifically, the requestor or a designated group of users will be given the opportunity to see, try, and comment upon prototypes at logical points.

As a prototype is refined, it approaches a point where it satisfies the solution requirements. At this point, either a paper copy of the SAPHIRE Change Design and Testing Checklist (Appendix B of INL/EXT-10-17915) or from the web version (see Figure 2) of the Checklist is completed. Completing this checklist will help assure that a standard list of coding issues have been addressed.