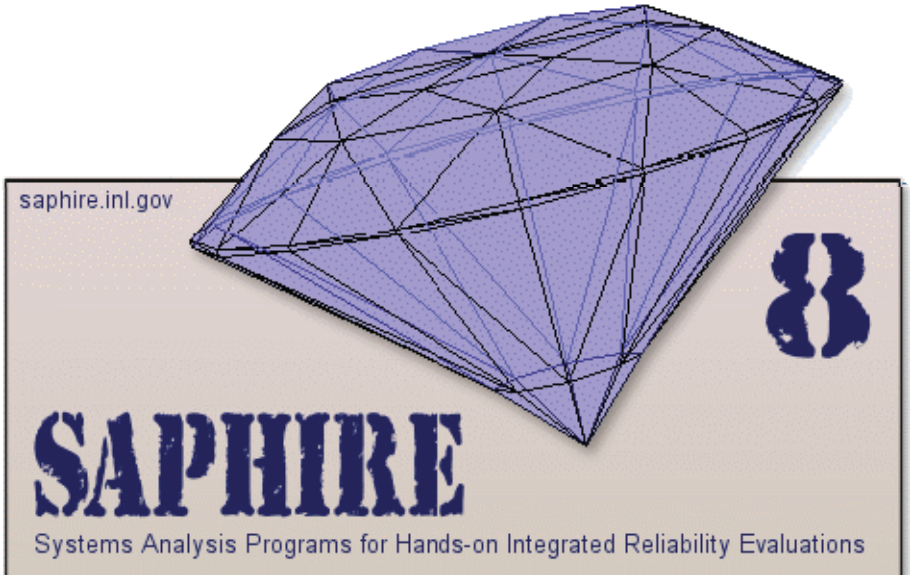
INL/EXT-09-17141

# Independent Verification and Validation of SAPHIRE 8 Software Configuration Management Plan

October 2009




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

Idaho National Laboratory

INL/EXT-09-17141

# Independent Verification and Validation of SAPHIRE 8 Software Configuration Management Plan

October 2009

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

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

# Table of Contents


1.0 Executive Summary ..... 1
2.0 Background Information ..... 2
3.0 Summary of Findings ..... 4
3.1 NUREG/BR-0167 Findings ..... 4
3.1.1 Section 2.5 Qualification Testing ..... 4
3.1.2 Section 3.2.3 Formal Peer Inspections ..... 4
3.1.3 Section 6.1 Concepts and Definitions ..... 5
3.1.4 Section 6.2 Baselines ..... 6
3.1.5 Section 6.3 Change Control ..... 6
3.1.6 Section 6.4 Status of Baselines and Changes ..... 7
3.1.7 Section 6.5 Software Development Library ..... 7
3.1.8 Section 6.6 Software, Access, and Media Control ..... 7
3.1.9 Section 6.8 Techniques and Tools ..... 7
3.1.10 Section 7.1 Nonconformance Reporting and Corrective Action ..... 8
3.2 Software Configuration Management Plan Document Findings ..... 8
4.0 IV&V Evaluation Checklist ..... 10

# 1.0 Executive Summary

The purpose of the Independent Verification and Validation (IV&V) role in the evaluation of the SAPHIRE configuration management is to assess the activities that results in the process of identifying and defining the baselines associated with the SAPHIRE software product; controlling the changes to baselines and release of baselines throughout the life cycle; recording and reporting the status of baselines and the proposed and actual changes to the baselines; and verifying the correctness and completeness of baselines.. The IV&V team began this endeavor after the software engineering and software development of SAPHIRE had already been in production.

The requirements for IV&V review were extracted primarily from the NUREG but also included an examination of best software engineering methods provided in the IEEE Standard for Software Verification and Validation. IV&V developed a checklist that mapped these requirements with these standards which was used in the evaluation. The evaluation criteria and the results of the assessment are identified in section 3 of this document.

Per the requirements and document outline provided in the SAPHIRE IV&V Plan, this report and all subsequent reports will be included as attachments and/or background evidence of the evaluation as well as the results of the assessment.

## 2.0 Background Information

NUREG/BR-0167, Software Quality Assurance Program and Guidelines, requires the development of configuration management activities that establish and maintain integrity of the software and its documentation as they evolve throughout the life cycle. The four major functions of configuration management include:

1. The identification and establishment of baselines.
2. Controlling both changes to baselines and the release of baselines.
3. Recording and reporting the status of baselines, change requests, and implemented changes.
4. Verifying, through auditing, the correctness and completeness of baselines prior to release.

This report provides an evaluation of the Software Configuration Management Plan. The Software Configuration Management Plan is intended to ensure the content and status of the software and documentation baselines are known at all times; the developer follows a written configuration management policy that has the following characteristics: (a) Responsibility for configuration management for each project is explicitly assigned. (b) Configuration management is implemented on products throughout the product's life cycle. (c) Configuration management is implemented for externally-deliverable products and for appropriate products used inside the organization. (d) All projects have a repository for storing key software engineering elements and associated configuration management records. (e) The software baselines and configuration management activities are audited on a regular basis; a group that is responsible for coordinating and implementing configuration management for the project exists or is established; adequate resources and budget for performing configuration management activities are provided; members of the configuration management group are trained in the objectives, procedures, and methods for performing their assigned activities; the configuration management activities are reviewed with the project manager on a regular basis, and to meet the contractual commitments prepared by the sponsor; the Nuclear Regulatory Commission.

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

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

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

To provide direction in the evaluation process, IV&V has developed a checklist to support the requirements for the SDLC. The Project Plan requirements used for the analysis of the Software Configuration Management Plan is contained in a checklist that is included in the SAPHIRE 8 Software Independent Verification and Validation Plan (INL/EXT-09-15649, Revision ID: 0, Effective Date: April 1, 2009). The evaluation criteria for the Software Configuration Management Plan have been extracted from the checklist contained in the "IV&V Plan" and included in section 4 of this report. A summary of the findings is provided in section 3.

# 3.0 Summary of Findings

An Independent Verification and Validation evaluation of the Software Configuration Management Plan Document ID: INL/EXT-09-16696 for SAPHIRE 8 was performed using the checklist contained in section 4.0. The checklist was extracted from the SAPHIRE 8 Software Independent Verification and Validation Plan Document ID: INL/EXT-09-15649. The following sections refer to the specific parts of the NUREG/BR-0167 Software Quality Assurance Program and Guidelines requirements the SAPHIRE 8 Software Configuration Management Plan failed to satisfy. Section 3.2 of the Summary of Findings lists minor findings between the Software Configuration Management Plan and section 2.7 of the Software Project Plan. Minor corrections for the Software Configuration Management Plan are also listed in 3.2 of the Summary of Findings.

## 3.1 NUREG/BR-0167 Findings

Refer to the SAPHIRE 8 Software Independent Verification and Validation Plan Document ID: INL/EXT-09-15649 for the substitution of Peer Review in place of Configuration Control Board (CCB).

### 3.1.1 Section 2.5 Qualification Testing

Fail "1" – refer to the Independent Verification and Validation Of SAPHIRE 8 Software Requirements, Document ID: INL/EXT-09-16789.

The qualification testing process is the set of activities associated with

1. Formally testing the implemented software, using test cases defined in the verification and validation documentation, against the baselined requirements defined in the software requirements documentation.
2. Reviewing and analyzing the test results to ensure that the implemented software meets requirements and that the software produces correct results for all test cases executed.

### 3.1.2 Section 3.2.3 Formal Peer Inspections

Fail

A formal peer inspection is a detailed examination of a product on a step-by-step or line-by-line basis. The purpose of conducting formal peer inspections is to find errors. The group that performs a peer inspection is composed of peers of the person who developed the product to be inspected. Peer inspections are objective approaches that have been proved very effective in verifying that products meet requirements.

For Level 1 software, require the developer to

1. Subject each intermediate product and final product of development and maintenance (i.e., all documentation, all code) to an internal peer inspection.
2. Make available to the sponsor the written procedure and the product standards that govern peer inspections.
3. Make available, if requested by the sponsor, records that document the results of all peer inspections.

### 3.1.3 Section 6.1 Concepts and Definitions

Fail "There are four major… 4" – refer to the Independent Verification and Validation Of SAPHIRE 8 Software Requirements, Document ID: INL/EXT-09-16789 and the Independent Verification and Validation Of SAPHIRE 8 Software Design and Interface Design, Document ID: INL/EXT-09-17069.

For a project to be successful, the developer and sponsor must establish and maintain integrity of the software and its documentation as they evolve throughout the life cycle. Because requirements, the design, the code, and the test environment can change significantly and often, it is essential that change be managed successfully. Briefly stated, configuration management is change management.

Fundamental to configuration management are the concepts of a baseline and change control. A baseline is a document or software that has been formally reviewed and agreed upon by the developer and sponsor, that thereafter serves as the basis for further development and that can be changed only through formal change control procedures. Change control is the process by which a change to a baseline is proposed, evaluated, approved or rejected, scheduled, and tracked.

There are four major functions of configuration management:

1. The identification and establishment of baselines.
2. Controlling both changes to baselines and the release of baselines.
3. Recording and reporting the status of baselines, change requests, and implemented changes.
4. Verifying, through auditing, the correctness and completeness of base lines prior to release.

For a software configuration management program to be successful, experience has shown that most of the following conditions exist:

1. The content and status of the software and documentation baselines are known at all times.
2. The developer follows a written configuration management policy that has the following characteristics:
   (a) Responsibility for configuration management for each project is explicitly assigned.
   (b) Configuration management is implemented on products throughout the product's life cycle.
   (c) Configuration management is implemented for externally-deliverable products and for appropriate products used inside the organization.
   (d) All projects have a repository for storing key software engineering elements and associated configuration management records.
   (e) The software baselines and configuration management activities are audited on a regular basis.
3. A group that is responsible for coordinating and implementing configuration management for the project exists or is established.
4. Adequate resources and budget for performing configuration management activities are provided.
5. Members of the configuration management group are trained in the objectives, procedures, and methods for performing their assigned activities.

6. The configuration management activities are reviewed with the project manager on a regular basis.

### 3.1.4 Section 6.2 Baselines

Fail "Establish the following baselines… 1".

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

Establish a naming or labeling system that:

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

Establish the following baselines that will be controlled by the sponsor's configuration control board (CCB):

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

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

### 3.1.5 Section 6.3 Change Control

Fail "3" – refer to the Independent Verification and Validation Of SAPHIRE 8 Software Requirements, Document ID: INL/EXT-09-16789 and the Independent Verification and Validation Of SAPHIRE 8 Software Design and Interface Design, Document ID: INL/EXT-09-17069.

Once a baseline has been established, changes to the baseline can be made only in accordance with formal change control procedures. To manage changes to baselines:

1. Establish a board (i.e., a configuration control board (CCB)) controlled by the sponsor project manager that has the authority for managing the software baselines and approving or rejecting proposed changes to them
2. Establish and follow a documented procedure for initiating, recording, reviewing, approving or rejecting, and tracking change requests for baselines.

3. Establish and follow a documented procedure for ensuring that all changes, especially those to the requirements and design, are appropriately reviewed for "ripple" effects and incorporated into all related activities.
4. Establish and follow a documented procedure to create and control the release of software baseline products.

### 3.1.6 Section 6.4 Status of Baselines and Changes

Pass

Track accurately the current status of baselines and changes throughout development and maintenance. To track status accurately:

1. Establish and follow a documented procedure to record the status of baselines and change requests.
2. Create and distribute to affected groups and individuals standard reports documenting the configuration management activities.

### 3.1.7 Section 6.5 Software Development Library

Pass

Require the developer to establish and maintain a software development library (SDL). An SDL is a controlled collection of software, documentation, and associated tools and procedures used to facilitate the orderly development and subsequent maintenance of software. The SDL contains the developmental configuration as part of its contents. An SDL provides storage of and controlled access to software and documentation in human-readable form, machine readable form, or both. The SDL may also contain management data pertinent to the software development project. The SDL becomes the repository for the software baselines when the product baseline and the operational baseline are established.

### 3.1.8 Section 6.6 Software, Access, and Media Control

Fail "3".

Require the developer to establish and maintain the facilities and procedures used to

1. Maintain, store, secure, and document controlled versions of the software throughout the life cycle. This may be implemented in the Software Development Library (SDL).
2. Permit authorized and prevent unauthorized access to the software and documentation.
3. Identify the media for each software product and the documentation required to store the media, including the copy and restore process.
4. Protect software physical media from unauthorized access on inadvertent damage or degradation throughout the life cycle.

### 3.1.9 Section 6.8 Techniques and Tools

Pass

Use a data base management system as a tool in tracking and reporting on proposed and actual changes to baselines. Often the data base of proposed and actual changes is integrated with the data base used to track and report on nonconformances and associated corrective action.

In addition, choose a software tool, often a part of the operating system utilities, to help manage the SDL.

### 3.1.10 Section 7.1 Nonconformance Reporting and Corrective Action

Pass

A nonconformance, often called a problem, discrepancy, fault, or error, is any failure of any document, code, data structure, or process to meet its requirements or standards. Corrective action is a general name for the process by which nonconformances are corrected, verified, and controlled.

Require the developer to establish and maintain a nonconformance reporting and corrective action system and associated procedures. The purpose of a nonconformance reporting and corrective action system is to report, analyze, correct, and verify nonconformances and collect information from which reports on the overall status of nonconformances can be made.

The need for a nonconformance reporting and corrective action system arises early in the software life cycle, as soon as the first documents and other products are developed. A nonconformance reporting and corrective action system should:

1. Define a nonconformance report form.
2. Identify the organization(s) and procedures for:
   (a) Analyzing the nonconformance.
   (b) Assigning priorities.
   (c) Communicating with the person who reported the nonconformance.
   (d) Correcting the nonconformance
   (e) Verifying the correction and/or the corrective action.
3. Track the status of the nonconformance and corrective action.
4. Produce management reports.

## 3.2 Software Configuration Management Plan Document Findings

1. **Section 1.2 Project Scope and Organization**. Sixth paragraph. Reference is made to *10 CFR 830 subpart A, "Nuclear Safety Management"*. 10 CFR 830 Subpart A is "Quality Assurance Requirements". Make corrections as necessary.
2. **Section 1.2 Project Scope and Organization**. Reference is made to "*LWP-13610"* as to the Software Owner responsibilities. Document LWP-13610 was not found. Make corrections as necessary.
3. **Section 1.2 Project Scope and Organization**. Tenth paragraph. Last sentence "*Currently, the SAPHIRE development does not reference or require a specific documented SQA process – thus the development falls under the INL SQA process*." SAPHIRE 8 has Software Quality Assurance Plan Document ID: INL/EXT-09-16697. Make corrections as necessary.
4. **Section 1.3 Configuration Management Approach**. First paragraph. Refer to the Software Project Plan, section 2.7 Configuration Management, paragraph two, last sentence and make corrections as necessary.
5. **Section 1.3 Configuration Management Approach**. Second paragraph. Refer to the Software Project Plan, section 2.7 Configuration Management, paragraph three, last sentence and make corrections as necessary. There is no Automated Testing section.

6. **Section 1.3 Configuration Management Approach**. Third paragraph. Refer to the Software Project Plan, section 2.7 Configuration Management, paragraph four and make corrections as necessary.
7. **Section 1.3 Configuration Management Approach**. Fourth paragraph. Refer to the Software Project Plan, section 2.7 Configuration Management, paragraph five, second sentence and make corrections as necessary.
8. **Section 1.3 Configuration Management Approach**. Seventh paragraph. The sentences "*The control library is kept on a server, where back-ups are regularly made. (Individual developers/programmers machines are periodically backed up as well)*." and "*A source code version control library requires that individual programmers "check-out" all files that they intend to modify. Prior to "check-in", programmers must explain any changes made. 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 ensures a methodical approach to tracking and releasing these changes*.", repeat information in paragraphs three, four and five. Make corrections as necessary.

# 4.0 IV&V Evaluation Checklist

| SOFTWARE CONFIGURATION MANAGEMENT | | |
|---|---|---|
| **Criteria Priority: 1** | **Does the Configuration Management approach/methodology identify, define and reference procedures used for establishing and maintaining project baselines? NUREG/BR-0167 Sections 2.5, 6.2, 6.4** | |
| Pass | X | Comments |
| Fail | | Section 1.2 Project Scope and Organization reference DOE Order 414.1C Quality Assurance, 10 CFR 830 Subpart A, Nuclear Safety Management, ASME NQA-1-2000 Quality Assurance Requirements for Nuclear Facility Applications, PDD-13610 Software Quality Assurance Program, LRD-13600 Software Quality Assurance, LWP-13620 Software Quality Assurance and NUREG/BR-0167 Software Quality Assurance Program and Guidelines.<br><br>Section 1.2 Project Scope and Organization, last paragraph, last sentence states "*SAPHIRE 8 will follow the requirements for Level 1 software defined in Section 1.2 of NUREG/BR-0167*".<br><br>Suggest referring to NUREG-BR-0167 Software Quality Assurance Program and Guidelines, section 6.2 Baselines for reference to all required information to be included in baselines. Section 1.3 does not specifically address the items included in the baselines.<br><br>Refer to section 3.0 Summary of Findings. |
| N/A | | |
| **Criteria Priority: 1** | **Does the Configuration Management approach/methodology identify, define and reference procedures used for establishing and performing change control? NUREG/BR-0167 Section 6** | |
| Pass | X | Comments |
| Fail | | Section 1.2 Project Scope and Organization reference DOE Order 414.1C Quality Assurance, 10 CFR 830 Subpart A, Nuclear Safety Management, ASME NQA-1-2000 Quality Assurance Requirements for Nuclear Facility Applications, PDD-13610 Software Quality Assurance Program, LRD-13600 Software Quality Assurance, LWP-13620 Software Quality Assurance and NUREG/BR-0167 Software Quality Assurance Program and Guidelines.<br><br>Section 1.2 Project Scope and Organization, last paragraph, last sentence states "*SAPHIRE 8 will follow the requirements for Level 1 software defined in Section 1.2 of NUREG/BR-0167*".<br><br>Suggest referring to NUREG-BR-0167 Software Quality Assurance Program and Guidelines, section 6.3 Change Control for reference to all required information to be included in the change control procedure.<br><br>Refer to section 3.0 Summary of Findings. |
| N/A | | |
| **Criteria Priority: 1** | **Does the Configuration Management approach/methodology identify, define and reference procedures used for implementation and release of changes? NUREG/BR-0167 Section 6** | |
| Pass | X | Comments |
| Fail | | Section 1.2 Project Scope and Organization reference DOE Order 414.1C Quality Assurance, 10 CFR 830 Subpart A, Nuclear Safety Management, ASME NQA-1-2000 Quality Assurance Requirements for Nuclear Facility Applications, PDD-13610 |
| N/A | | |

| | | |
|---|---|---|
| | | Software Quality Assurance Program, LRD-13600 Software Quality Assurance, LWP-13620 Software Quality Assurance and NUREG/BR-0167 Software Quality Assurance Program and Guidelines.<br><br>Section 1.2 Project Scope and Organization, last paragraph, last sentence states "*SAPHIRE 8 will follow the requirements for Level 1 software defined in Section 1.2 of NUREG/BR-0167*".<br><br>Refer to section 3.0 Summary of Findings. |
| **Criteria Priority: 1** | **Does the Configuration Management approach/methodology identify, define and reference procedures used for code, access, and media controls?<br>NUREG/BR-0167 Section 6** | |
| Pass | X | Comments |
| Fail | | Section 1.2 Project Scope and Organization reference DOE Order 414.1C Quality Assurance, 10 CFR 830 Subpart A, Nuclear Safety Management, ASME NQA-1-2000 Quality Assurance Requirements for Nuclear Facility Applications, PDD-13610 Software Quality Assurance Program, LRD-13600 Software Quality Assurance, LWP-13620 Software Quality Assurance and NUREG/BR-0167 Software Quality Assurance Program and Guidelines.<br><br>Section 1.2 Project Scope and Organization, last paragraph, last sentence states "*SAPHIRE 8 will follow the requirements for Level 1 software defined in Section 1.2 of NUREG/BR-0167*".<br><br>Suggest referring to NUREG-BR-0167 Software Quality Assurance Program and Guidelines, section 6.6 Software, Access, and Media Control for reference to all required information to be included in the software, access, and media control procedure.<br><br>Refer to section 3.0 Summary of Findings. |
| N/A | | |
| **Criteria Priority: 1** | **Does the Configuration Management approach/methodology identify, define and reference procedures for the use, access, and maintenance of the software development library?<br>NUREG/BR-0167 Section 6** | |
| Pass | X | Comments |
| Fail | | Section 1.2 Project Scope and Organization reference NUREG/BR-0167 Software Quality Assurance Program and Guidelines.<br><br>Refer to section 3.0 Summary of Findings. |
| N/A | | |
| **Criteria Priority: 1** | **Are all nonconformance items under CM Control?<br>NUREG/BR-0167 Section 6 and 7** | |
| Pass | X | Comments |
| Fail | | Refer to section 3.0 Summary of Findings. |
| N/A | | |
| **Criteria Priority: 1** | **Are the monthly progress reports under configuration management control?<br>NUREG/BR-0167 Section 6** | |
| Pass | | Comments |
| Fail | X | Refer to section 3.0 Summary of Findings. |
| N/A | | |
| **Criteria Priority: 1** | **Are peer reviews and structured walkthrough documents/completed forms under configuration control?<br>NUREG/BR-0167 Section 3.2.3** | |
| Pass | | Comments |
| Fail | X | Refer to section 3.0 Summary of Findings. |
| N/A | | |

| | | |
|---|---|---|
| **Criteria Priority: 1** | **Does the developer follow a written configuration management policy/methodology? NUREG/BR-0167 Section 6.1** | |
| Pass | X | Comments |
| Fail | | Refer to section 3.0 Summary of Findings. |
| N/A | | |
| **Criteria Priority: 1** | **Are baseline documents for planning, managing and building the system (software) established and controlled (Explicitly identify project baselines for software products (source code, test cases, software specifications (standards & procedures) needed to establish & maintain stability of software activities? NUREG/BR-0167 Section 6.2** | |
| Pass | X | Comments |
| Fail | | Refer to section 3.0 Summary of Findings. |
| N/A | | |
| **Criteria Priority: 1** | **Have a naming / labeling system that: uniquely identifies all project entities (documents, software elements, and test cases), changes by revision or version (and under CM Control), unique identification of configuration/version of revised software for use? NUREG/BR-0167 Section 6.2** | |
| Pass | X | Comments |
| Fail | | Refer to section 3.0 Summary of Findings. |
| N/A | | |
| **Criteria Priority: 1** | **Are baseline documents for planning, managing and building the system (software) established and controlled? NUREG/BR-0167 Section 6.2** | |
| Pass | X | Comments |
| Fail | | Refer to section 3.0 Summary of Findings. |
| N/A | | |