

Enclosure 5

MFN 07-384

**ESBWR Licensing Topical Report –
Software Management Plan, Rev. 2 –**

NEDO-33226

Non-Proprietary Version



**GE Energy
Nuclear**

3901 Castle Hayne Rd
Wilmington, NC 28401

NEDO-33226
Revision 2
Class I
DRF#0000-0051-3897
July 2007

LICENSING TOPICAL REPORT

ESBWR I&C SOFTWARE MANAGEMENT PLAN

Copyright, 2007 General Electric Company

Important Notice Regarding Contents of this Report
Please read carefully

The information contained in this document is furnished as reference to the NRC Staff for the purpose of obtaining NRC approval of the certification and implementation. The only undertakings of General Electric Company with respect to information in this document are contained in contracts between General Electric Company and participating utilities, and nothing contained in this document shall be construed as changing those contracts. The use of this information by anyone other than that for which it is intended is not authorized; and with respect to any unauthorized use, General Electric Company makes no representation or warranty, and assumes no liability as to the completeness, accuracy, or usefulness of the information contained in this document.

NON-PROPRIETARY INFORMATION NOTICE

This document is the non-proprietary version of NEDO-33226, Rev 1, and thus, has the proprietary information removed. Portions of this document that have been removed are indicated by open and closed double brackets, as shown here [[]].

Copyright, 2007 General Electric Company

II

ii

1.0 INTRODUCTION

1.1 Overview

The Software Management Plan (SMP) includes the key planning documents for the Instrumentation and Controls (I&C) design team and governs the design and development activities for the Digital Computer-Based I&C software for the ESBWR.

1.2 Purpose and Scope

The scope of the SMP includes software products with the software classifications of Software Class Q, N3, and N2. The definitions for software classifications are defined in Appendix C. Unless otherwise specified, non-safety systems are referenced as Software Class N in the SMP.

The software plans are identified in the ESBWR Man-Machine Interface (MMI) System and Human Factor & Engineering (HFE) Implementation Plan [2.1(1)]. The software plans included in this SMP document, referred to as the Software Management Plan, are:

- | | |
|---|---------------|
| 1. Software Development Plan (SDP) | [Section 5.0] |
| 2. Software Integration Plan (SIntP) | [Section 6.0] |
| 3. Software Installation Plan (SIP) | [Section 7.0] |
| 4. Software Operation and Maintenance Plan (SOMP) | [Section 8.0] |
| 5. Software Training Plan (STrngP) | [Section 9.0] |

The ESBWR I&C Software Quality Assurance Plan (SQAP) [2.3(1)], herein referred to as SQAP, includes the software plans used by the Quality Assurance (QA) and the Software Project Engineering (SPE) organizations, governing the same I&C software scope identified in the MMIS/HFE IP.

- | | |
|---|-----------------------|
| 1. Software Verification & Validation Plan (SVVP) | [2.3(1) Section 7.0] |
| 2. Software Safety Plan (SSP) | [2.3(1) Section 9.0] |
| 3. Software Configuration Management Plan (SCMP) | [2.3(1) Section 10.0] |

Together, the SMP and the SQAP include all the software plans identified in Reference 2.1(1) and conform to the guidance provided by NUREG 0800, Standard Review Plan [2.2.1].

This SMP shall be in force during all phases of the software life cycle.

The applicable Software Products (software and firmware) covered in this SMP encompass all I&C systems, as specifically defined in the MMIS/HFE IP [2.1(1)] (Subsection 1.2.4 only), which perform the monitoring, control, alarming, and protection functions associated with all modes of ESBWR plant normal operation (i.e., startup, shutdown, standby, power operation, and refueling) as well as off-normal, emergency, and accident conditions.

1.3 Acronyms, Abbreviations, and Definitions

Acronyms and abbreviations are defined in Appendix B. Definitions for terms used in this SMP

are supplied in Appendix C.

2.0 APPLICABLE DOCUMENTS

Applicable documents include supporting documents, supplemental documents, codes and standards, and are identified in this section. Supporting documents provide the input requirements to this SMP. Supplemental documents are used in conjunction with this SMP. Codes and standards that are applicable are also identified in this SMP.

2.1 Supporting Documents

The following supporting documents were used as the controlling input documents in the development of this SMP. These documents form the design basis for the activities stated in this SMP. This document governs, in the event of any differences noted between this SMP and the ESBWR Composite Design Specification [2.1(2)].

1. ESBWR Man-Machine Interface System and HFE Implementation Plan (MMIS/HFE IP), NEDO-33217, Rev. 3.
2. ESBWR Composite Design Specification (A11-5299), 26A6007, Rev. 0.
3. ESBWR Composite Design Specification Standard Review Plans and Regulatory Guides (A11-5299), 26A6007AB, Rev. 3.
4. ESBWR Composite Design Specification Industry Codes and Standards (A11-5299), 26A6007AC, Rev. 2.
5. ESBWR DCD, Chapter 7, I&C Systems, 26A6642AW Rev. 4.
6. ESBWR DCD, Chapter 15, Safety Analysis, 26A6642BP, Rev. 3.

2.2 Codes and Standards

The following codes and standards are used in conjunction with this SMP.

2.2.1 NUREG

The following codes and standards are applicable to the activities specified within this SMP. This SMP conforms to planning requirements of these codes and standards except as explicitly noted in Appendix A.

1. NUREG 0800, Standard Review Plan (SRP), Chapter 7, Branch Technical Position (BTP).
2. HICB-14 R4, Guidance on Software Reviews for Digital Computer-Based I&C Systems.

2.2.2 Code of Federal Regulations (CFR)

1. 10 CFR50, Appendix – B, Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants.

2.2.3 U.S. Nuclear Regulatory Commission (NRC) Regulatory Guides (RG)

The following codes and standards are applicable to the activities specified within this SMP. This SMP conforms to planning requirements of these codes and standards except as explicitly noted in Appendix A.

1. RG 1.152-2006 – Criteria for Use of Computers in Safety Systems of Nuclear Power Plants.
2. RG 1.168-2004 – Verification, Validation, Reviews, and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
3. RG 1.169-1997 – Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
4. RG-1.170-1997- Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
5. RG-1.171-1997 – Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
6. RG 1.172-1997 – Software Requirements Specifications for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.
7. RG 1.173-1997 – Developing Software Life cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.

2.2.4 Institute of Electrical and Electronic Engineers (IEEE) Standards

The following codes and standards are applicable to the activities specified within this SMP. This SMP conforms to planning requirements of these codes and standards except as explicitly noted in Appendix A.

Where these IEEE Standards provide recommended implementation techniques and methods, this SMP makes specific commitments only to those requirements restated herein. The ESBWR Project Work Plans shall capture the detailed implementation attributes in accordance with EOP 25-5.00 [2.3(2a)]. Future exceptions or deviations from the recommendations specified in the IEEE standards shall require management approval as defined in the SQAP [2.3(1)] and this SMP, and are potentially subject to NRC notification. The NRC notification process is addressed in the MMIS/HFE Implementation Plan [2.1(1)].

1. IEEE 7-4.3.2-2003 IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations.
2. IEEE 1012-1998 IEEE Standard for Software Verification and Validation.
3. IEEE 1028-1997 IEEE Standard for Software Reviews Description.
4. IEEE 828-1990 IEEE Standard for Software Configuration Management Plans.
5. IEEE-1042-1987 IEEE Guide to Software Configuration Management Description.
6. IEEE-829-1983 IEEE Standard for Software Test Documentation.
7. IEEE-1008-1987 IEEE Standard for Software Unit Testing.

| Document Title | Document Number |
|--|----------------------------|
| 4. GE Energy, “ESBWR Cyber Security Program Plan”, Class I (Non-Proprietary), and ESBWR Cyber Security Program Plan”, Class III (Proprietary). | NEDO-33295 NEDO-33495-P |
| 5. ESBWR HFE Training Development Implementation Plan | NEDO-33275 |
| 6. IEEE Standard Glossary of Software Engineering Terminology | IEEE 610.12-1990 |
| 7. Guidelines on Evaluation and Acceptance of Commercial Grade Digital Equipment in Nuclear Safety Applications | EPRI TR-106439 |

2.4 Additional IEEE Standard Guidance

The following IEEE Standards provide additional guidance for the implementation activities. Conformance of this SMP to these activities has been evaluated. Selected sections/topics from these IEEE Standards are excluded from commitment because they either provide conflicting requirements with other Standards or the level of detail is not appropriate for this SMP. Clarifications and justifications for such exclusions are provided in Appendix A.

1. IEEE-603-1998 – IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations.
2. IEEE-730-2002 – IEEE Standard for Software Quality Assurance Plans.
3. IEEE-1016-1998 – IEEE Recommended Practice for Software Design Descriptions.
4. IEEE-1058.1-1987 – IEEE Standard for Software Project Management Plans.
5. IEEE 1219-1998 – IEEE Standard for Software Maintenance.
6. IEEE 1228-1994 – IEEE Standard for Software Safety Plans.
7. IEEE 12207-1996 – IEEE/EIA Standard for Software Life Cycle Processes.

2.5 International Standards

1. ISO 9001:2000, Quality Management Systems-Requirements.

3.0 SOFTWARE MANAGEMENT PLAN

3.1 Purpose and Scope

The purpose of the SMP is to establish the managerial process and the technical direction for the design and development activities of the Digital Computer-Based I&C Software within the scope of the MMIS/HFE IP [2.1(1)].

3.2 Organization

The organization is established to address the control of software management and to ensure that independence is maintained between the design organization and the quality assurance, software safety, and Verification and Validation (V&V) organizations. The organization is shown in Figure 3.1.

This section describes the following ESBWR organization functions:

1. I&C and Electrical Systems Engineering.
2. Software Project Engineering (SPE).
3. Configuration Management. (HFE).
4. Project Management (i.e. Project Control).
5. Training.

3.2.1 I&C and Electrical Systems Engineering

The I&C software development organization is comprised of the GEEN I&C and Electrical Systems Engineering organization and the (GEEN and non-GEEN) software products vendor organization. The GE I&C and Electrical System Engineering (I&C/ESE) Organization is comprised of the I&C and Electrical Systems Engineering Manager (I&C Manager), the platform Technical Project Engineers (TPEs), and the responsible I&C/ESE. This organization implements the activities defined in the SMP.

The I&C Manager is responsible for overall performance and schedule of the software development effort, including work flow to the system TPEs, system engineers, and software products vendors. The platform TPEs are responsible for day to day management, coordination, and scheduling of the system design and software development effort and are responsible for interfacing with the system engineers and software product vendors. The TPEs are also responsible for providing status reports to the I&C Manager.

The I&C/ESE engineer is responsible for the design and development of the software products. The I&C/ESE is responsible for review, and for confirming that the design documentation and outputs produced by the software products vendors meets the technical requirements specified in the contract/purchase order.

The software product vendors shall produce the software described in this SMP. The vendors may be internal or external to GEEN. The vendors shall be organized such that a single Point of

Contact (POC) is assigned the responsibility of interfacing with the TPE. Alternative POCs shall be assigned to take over the duties of the POC whenever the Primary POC is unavailable. The POC and alternative POCs shall be determined by the hardware/software vendor organization and may be any individual within the organization who is qualified to act as the organization's agent. Software developed by the vendors shall be in accordance with this SMP and the SQAP [2.3(1)].

[[

]]

3.2.2 Software Project Engineering

The SPE is independent of the I&C/ESE organization to ensure organizational freedom to perform the Quality tasks without undue pressure or conflict of interest related to budget or schedule.

The SPE organization, which is comprised of the following teams, is responsible for executing the quality tasks as described in the SQAP [2.3(1)]:

- Independent Verification and Validation Team (IVVT)
- Software Safety Analysis Team (SST)
- Baseline Review Team (BRT)

The SPE organization is described in the SQAP [2.3(1)] Subsection 3.3.5. The Simulation Assisted Engineering (SAE) and HFE teams do not perform the quality tasks and are not described in the SQAP.

3.2.3 Training

Training requirements are addressed in Section 4.5 of this SMP.

3.2.4 Configuration Management Manager

The Configuration Management Manager (CMM) has the overall responsibility and authority for the Configuration Management System (CMS), herein referred to as Product Data Management System (PDMS). The CMM responsibilities are addressed in the SCMP [2.3(1) Section 10.0].

3.2.5 Software Quality Assurance Manager

The SQA Manager interfaces with the SPE Manager and has the overall responsibility and authority for the SQA program. The SQA Manager responsibilities are addressed in the SQAP [2.3(1)].

3.2.6 Project Management Team

The technical management of software products is the responsibility of the TPEs. The Project Management Team (PMT) is responsible for the commercial aspects of the project. A commercial Project Manager (PM) shall be assigned to oversee each of the projects, and shall be responsible for delivering the commitments of a Purchase Order and/or Sales Contract to the Licensee.

The following activities are included:

1. Project work planning.
2. Development and maintenance of the integrated project schedule. The TPEs shall provide task inputs and support for this activity.
3. Update of the integrated schedule to show that project tasks are completely and accurately reflected.
4. Assignment of project resources and skill sets to support the project needs.

5. Preparation of project progress reports.
6. Project risk management assessment.
7. Project budgeting.
8. Engineering procurement and/or fabrication.
9. Communication with Licensee and vendors.

3.3 Organizational Boundaries and Interfaces

The SMP and SQAP specify the organizational structures for the I&C and Electrical Systems Engineering and SPE. This includes boundaries and relationships with the external and internal organizations. The PM provides the Licensee and vendor interface with the I&C/ESE and SPE organization. [[

]]

3.4 Organizational Responsibilities

Organizational responsibilities are defined in the following Sections:

3.4.1 New Units Engineering Manager

The New Unit Engineering Manager is responsible for the New Units Engineering organization. The ESBWR Engineering, Configuration Management, and SPE managers report to the New Units Engineering Manager.

3.4.2 ESBWR Engineering Manager

The ESBWR Engineering Manager is responsible for directing the engineering activities of the ESBWR Project. The ESBWR Engineering Manager reports to the New Units Engineering Manager.

3.4.3 I&C and Electrical Systems Engineering Manager

The I&C Manager is responsible for directing the engineering work of the I&C and Electrical Systems Engineering organization. The functional leads for various I&C/ESE functions report to the I&C and Electrical Systems Engineering Manager. The I&C/ESE Manager reports to the ESBWR Engineering Manager.

3.4.4 SPE Manager

The SPE Manager is responsible for the software quality tasks during the design and development of the software product. The SPE Manager reports to the I&C New Unit Engineering Manager.

3.4.5 SQA Manager

The SQA Manager, who interfaces with the SPE Manger, has the overall responsibility and authority of SQA Program. The SQA Manager reports to the Quality General Manager.

3.4.6 Training Services Lead (TSL)

The TSL is responsible for organizing the overall training process, including scheduling, budgeting, and resource allocation. The TSL reports to the Plant Performance and Optimization Manager.

3.4.7 Configuration Management Manager (CMM)

The CMM is responsible for the configuration management of the ESBWR project, including software products. The CMM reports to the New Units Engineering Manager.

3.4.8 Technical Project Engineer (TPE)

The TPEs have technical responsibility for the software tasks related to software or a group of software products. The TPE's report to the I&C/ Electrical Systems Engineering Manager.

3.5 Software Management Plan Change Control Process

The SMP is applicable for the entire life cycle of the software product. It is anticipated that the software development cycle shall evolve with changes in software development technology. It is acceptable to revise the SMP to improve quality. The change control process is described in the SQAP [2.3(1)].

The SMP is a controlled document under configuration control in accordance with the SCMP [2.3(1) Section 10.0]. [[

]] If a change to the SMP is warranted, one of the SQA activities shall determine if NRC notification is required and shall track the notification process as defined by the MMIS HFE IP [2.1(1)].

Changes to the SMP require approval of the I&C Manager and SPE Manager or designated appointees. [[

]] If changes to the SMP are made, the SQA Manager must document an evaluation showing that previously completed projects do not have to be reopened to implement the SMP changes. When changes are made to the SMP, requirements traceability will be maintained and verified.

4.0 MANAGEMENT PROCESS

4.1 Management Priorities, Monitoring, and Control

The objective of project management is to coordinate the development of project deliverables and to assure that the deliverables meet the Licensee expectations for nuclear safety, quality, cost, and schedule. The key elements for a successful project delivery by the project management are:

- Integrity - Integrity for all aspects of project performance is practiced at all times.
- Quality – Compliance with the software development and quality assurance process defined in the SMP, the SQAP [2.3.1(14)], and the applicable industry codes and standards.
- Occupational Safety - Safe work habits are practiced at all times.
- Outputs - Deliverables meet the quality, schedule, and budget requirements as specified by the project work plans.

[[

]]

The key management processes are:

1. Project Initiation.
 2. Project Planning and Scheduling.
 3. Project Execution.
 4. Project Controls.
- Post-Delivery Closeout.

4.1.1 Project Initiation

Project initiation begins after the contract has been awarded or an internal project is authorized. A preliminary schedule is developed, which considers project resource availability, and is consistent with the approved project work scope and budget.

4.1.2 Project Planning and Scheduling

[[

]] Project planning kickoff meetings with the Licensee are conducted to confirm that the contractual requirements are implemented according to the risk management work scope in the PWP. Risk Assessment and Risk Management are performed, as defined in Section 4.2, and a plan for risk abatement is initiated and documented in the PWP.

The PWP is prepared to identify the associated work scope, design inputs and outputs, deliverables, and QA requirements, as described in the SQAP [2.3(1)]. Timing for these activities shall be consistent with the integrated project schedule.

[[

]]

The PWP is updated as changes occur in the work scope to reflect the current project status as determined by the PM.

The Software Development Plan, Section 5.0, describes the life cycle phases used in the design and development of the software products and design outputs for each life cycle phase. This includes Software Safety Analysis (SSA) report, Independent Verification and Validation (IV&V) report, and configuration control of the design outputs for each life cycle phase.

The phase Baseline Reviews (BRs) shall be conducted for each software product or logical group of projects at the end of each life cycle phase as described in the SQAP [2.3(1)]. The reports are prepared in accordance with the SQAP [2.3(1)].

4.1.3 Project Execution

These processes are performed to complete the work defined in the PWP. The objective is to carry out planned activities and processes using resources to meet the project objectives. Project Execution includes the following:

1. Initiating of material requisitions for services and materials.
2. Conducting project kickoff meetings with all interfacing organizations (i.e., Licensee and vendors). The frequency of the project meeting is determined by the PM and shall be conducted with the internal organization, such as engineering and SPE , and the external organization, Licensee or vendor.
3. Monitoring software product development progress.
4. Identifying critical path items/activities.
5. Establishing/reviewing milestone dates for these items/activities.
6. Identifying and adjusting manpower and resource levels.
7. Identifying internal and vendor performance problems as early as possible.

4.1.4 Project Controls

Project Controls activities include measurement and monitoring of project execution. The metrics that are integral to the Workforce Planning and Scheduling Tools are applied by the PM so that corrective action can be taken when necessary to adjust for schedule delays, unexpected changes in work scope, or quality issues stemming from design team and vendor performance challenges.

[[

]]

Project performance is monitored using:

1. Business financial tracking tools.

2. Managerial/Project review. The Managerial/Project review is used to assess the execution of the project.

4.1.4.1 Frequency of project review

Frequency of project review should be commensurate with the complexity of the project. Project status shall be presented during project reviews including reporting of earned value. Frequency and type of project reviews are described in the PWP.

4.1.4.2 Progress reports

Progress reports, which detail progress and status on a regular basis to the Licensee, are prepared by the PM. The frequency of the progress report is specified in the contract and in the PWP.

4.1.5 Post-Delivery Closeout

The objective of Post-Delivery Closeout is to finalize the product and to complete the delivery in accordance with the contract. The closure of project paperwork, including, Design Record File or (DRFs), closeout of vendor activities, including, vendors submit required documentation to GE, and turn over of the project to the Licensee, including, transfer of SQA activities to Licensees. The following shall also be performed during post-delivery closeout.

4.1.5.1 Project Deliverables

The project deliverables include a combination of hardware, software, design documents, and supporting documents such as test and analysis reports. The project deliverables are identified in the Licensee contracts. The PWP also identifies the milestone dates associated with the project delivery.

4.1.5.2 Software Developed by Vendors

A software product developed by vendors shall conform to the requirements outlined in this SMP and the SQAP [2.3(1)]. Class Q software is approved by the I&C Manager and the SPE Manager. Class Q software is audited by the SQA team.

[[

]]

4.2 Risk Management

Risk Management is the process of identifying, controlling, and mitigating events that may affect the project cost and schedule, to ensure that nuclear safety and security are maintained. [[

]]

4.3 Security

4.3.1 Physical Security

The design and development of software products shall be performed in a secured environment in accordance with Regulatory Guide 1.152. The test facility shall use controlled access (e.g., badge access). Access shall be limited to test personnel who are responsible for the software product being developed and tested. If multiple tests are being performed at the same time, special care shall be taken and a boundary shall be set up to assure that the test equipment, test documentation, and production hardware are not compromised.

4.3.2 Software Security

Software Security is an important consideration during the development of software products. Methods and procedures are applied to prevent contamination of the software products with viruses, Trojan horses, or other nefarious intrusions throughout the software life cycle of the software product. Design requirements shall ensure that all software be protected from internal/external attacks. Design Requirements shall also include prevention of unauthorized and inappropriate access.

Safeguards shall be in place to prevent any software product from becoming contaminated by these threats and intrusions. The safeguards shall include the following:

1. The development platform for the software product shall be isolated from external networks.
2. Cyber security procedures for software product development shall be followed.
3. The I&C Design Engineers shall be trained in the cyber security requirements and procedures.

The cyber security program is further defined by a separate Licensing Topical Report, NEDO-33295, ESBWR Cyber Security Program Plan [2.3(4)].

4.4 Methods and Tools for Project Management

4.4.1 Methods

Project management methods are addressed in Sections 4.1 and 4.2 of this SMP.

4.4.2 Tools

The Project Manager shall specify approve which tools in the PWP are required for efficient performance of the project. To assure efficient and effective execution of the software, the product GEEN Project Team shall be provided with the tools for project management such as:

1. Computers and/or notebooks/laptops [[]], which can support the software tools needed to aid the project management activities.
2. High speed printers, copiers, and scanners.

3. Software programs such as Microsoft® Word, Excel, Outlook and Adobe® Acrobat, which are widely and commonly used to assure efficient communication with Licensee and/or vendors.

Workforce planning and scheduling tool, e.g., Primavera P3, which allows:

1. The PMs to plan activities, and develop and maintain project schedules to track project progress.
2. The assignment of resources to ensure that resource requirements can be met by available resources with appropriate resource skill-sets to support the project.
3. The ability to ensure that the resource requirements can be met with appropriate resource skill-sets to support the project.

Product Data Management System, e.g., eMatrix, is:

1. A computer-based data system that stores, retrieves, and reports data relevant to the engineering definition of products and services offered and provided to the Licensee.
2. The official configuration control system for engineering controlled documents.

[[

]]

4.5 Training and Qualification

[[

]] The Engineering and Project training is performed either by classroom or individual study of the required EOPs and P&Ps. This SMP, SQAP [2.3(1)], and applicable tools are needed to support the design work. [[

]]

In addition, project requirements mandate that personnel receive training on processes, procedures, and tools, as required to support the specific project. The use of such tools shall be documented in the PWP. [[

]]

4.6 Budget

It is imperative that all costs and commitments are considered when analysis of project costs is being performed. This is performed in order to achieve the correct cost budget for the project. This may result in a possible adjustment to the current estimate at completion to support the project commitments, especially those related to safety and quality of the software products.

[[

]]. The PMT is independent of the Engineering Teams that are responsible for the design and quality assurance work on the project. The software product activities are planned and scheduled, and a cost estimate is developed based upon the software product Work Breakdown Structure (WBS). This enables budgets to be estimated for the project. The specific budgetary activities are:

1. Accurately allocate resources to each project organization, including vendors internal and external to the project organization.
2. Assign resources to each project organization to maintain financial independence from each other.

The PM is responsible for generating the project task charge numbers based on the identified and scheduled activities, so that expenditure can be monitored at the task level. Unique charge numbers are generated for each activity or a set of similar activities. For example, the charge numbers assigned for software implementation work is different from IV&V and SSA activities.

Any expenses incurred by the project shall be charged to appropriate charge numbers. The expenses include, but are not limited to labor, including GEEN employees and contractors, travel and living expenses, external contract labor, and purchased material.

A financial review shall be conducted on a quarterly basis with the New Plant Project (NPP) General Manager to assure that the costs incurred are consistent with the approved budgets. If the project estimate at completion is different from the approved budgeted cost, the project cost budgets will need to be adjusted to match the cost at completion. It is imperative that all costs and commitments are considered when analyzing project costs in order to achieve the correct cost budget for the project. This may result in a possible adjustment to the current estimate at completion to support the project commitments, especially those related to safety and quality of the software products.

)

5.0 SOFTWARE DEVELOPMENT PLAN

5.1 Introduction

The Software Development Plan (SDP) describes the plan for technical project development of the I&C software, which performs the monitoring, control, and protection functions for all modes of plant operation.

5.2 Purpose and Scope

The SDP describes the software engineering development process for each phase of the software products life cycle process. The phases include Planning, Requirements, Design, Implementation, Test, Installation, Operations & Maintenance (O&M), and Retirement. The SDP also addresses the preparation, execution, and documentation of software testing for software products. The SDP conforms to RG 1.173 [2.2.3(7)] and IEEE 1074 [2.2.4(9)], except as specified in Appendix A.

The purpose of the SDP is to:

1. Establish the standards, methods, tools, and procedures for the software design and development process.
2. Define the activities performed for each phase of the software development.
3. Define how requirements are traced to lower levels of the engineering from planning phase to test phase.
4. Specify how the safety-related requirements are documented, evaluated, reviewed, verified, and tested during the design process to minimize unknown, unreliable, and abnormal conditions.
5. Describe the organization and responsibilities of individuals or groups involved in the various V&V and review activities.
6. Provide a structure for test and review guidance for software functional testing during the software life cycle.
7. Provide the requirements and guidelines necessary to prepare, execute, and document software tests.
8. Address software test documentation.
9. Address metrics to include error tracking and resolution.

5.3 Organization of Software Life Cycle Process

Software engineering is a set of formal elements (methods, tools, documents, practices, standards, and procedures) applied during each phase of the software life cycle. The software life cycle phases defined in this SMP conform to and are based on RG 1.152 [2.2.3(1)], RG 1.173 [2.2.3(7)], and IEEE 1074 [2.2.4(9)]. A well-defined software engineering process implemented in a traceable, planned, and orderly manner is important for the development and maintenance of high quality software. [[

]] The software life cycle phases are described as follows:

Planning Phase – In this phase, the definition of the project scope, methodologies, and resources needed to develop and maintain the deliverable software is determined. The planning activities include evaluation of system and Licensee requirements, identification of resources, and development of schedule projections and risk assessments. The Planning Phase Baseline Review Report (BRR) documents successful completion of this phase.

Requirements Phase – In this phase, the definition of the detailed functional and performance requirements, design constraints, and validation criteria is determined. The Requirements Phase BRR documents successful completion of this phase.

Design Phase – This is the phase of the process that transforms requirements into architecture and a detailed representation of software. The Design Phase BRR documents successful completion of this phase.

Implementation Phase – During the implementation phase, the software design is transformed into software source or application codes. The implementation phase activities also include software code review and software functional tests. A Software functional test is conducted to validate the software source or application codes using a structured test approach. Software-software and software-hardware integration is performed during software functional testing. Typically, prototype hardware is used at this time. Implementation Phase BRR documents successful completion of this phase

Test Phase – This phase includes the software validation testing, which tests for potential defects (errors). The results are documented in the software validation test report. The Test Phase BRR documents successful completion of this phase

Installation Phase – This phase includes all activities associated with the installation of the validated software product into the target environment up through final product installation at the plant.

[[

]]

The Site Acceptance Test (SAT) integrated systems test is performed at the Licensee site. The results are documented in the Site Acceptance Test Report(s).

The Installation Phase BRR documents successful completion of this phase.

Operations & Maintenance Phase – This phase involves the functional and operational life of the software product(s). It includes the operation, maintenance, calibration, surveillance, and other processes associated with the use of the system. Application of the processes is based on

data, documentation, and procedures provided with each system in the O&M manual. The Maintenance section of the O&M manual includes procedures to maintain and resolve any operational anomalies. [[

]]

Retirement Phase – In the retirement lifecycle phase, the effect of replacing or removing the existing software product from the operating environment must be addressed.

The activities which address these effects should include:

- User notification effect on existing software products that are to remain operational in the operating environment
- Effect on existing software products that are to remain operational in the operating environment
- Disposition of the retired software product including security disposition. This includes:
 - Deactivation
 - Deletion or the removal of the software product from the operating environment
 - Operational comparison of the new and old software products
 - Any documentation activities, including archiving of records

5.4 Methods and Tools

5.4.1 Methods

The following methods are used to support the design and development of the software product.

5.4.2 Configuration Management and Change Control

[[

]] A discrepancy or deficient condition detected in a CI shall be resolved in accordance with the Change Control process described in the SCMP [2.3(1) Section 10.0].

5.4.3 Independent Verification

Independent Verification and Validation (IV&V) shall be conducted. [[

]]

5.4.4 Testing

Testing is conducted to assure the correctness of constructed code and completeness of requirements specified in the Requirements and Design Phase documents. [[

]]

5.4.5 Software Safety Analysis

A Software Safety Analysis (SSA) shall be performed to ensure the safety of the Class Q and N3 software. Safety is the most important consideration for the safety related I&C, taking precedence over budget and schedule. A SSA for software shall be performed according to the SSP [2.3(1) Section 9.0]. [[

]]

5.4.6 Baseline Review

A baseline review is performed for each software product at the completion of each software life cycle phase. This is consistent with the SSP [2.3(1) Section 9.0]. [[

]]

5.4.7 Deferred Design Verification

Conditional release of a design document may be permissible for cases where a design, or portion(s) of a design, must be released prior to completion of independent verification. Independent Verification is conducted in accordance with the SVVP [2.3(1), section 7.0]. [[

]]

5.5 Tools

Specific tools that are required for the project, which may include but are not limited to materials, prototypes, hardware, simulators, emulator, and support software shall be documented [[]]

5.5.1 Support Software

Support software is considered a tool used to aid the development of the software product throughout the software development process. [[]]

5.5.2 Requirements Traceability Matrix

A Requirements Traceability Matrix shall be prepared for Software Class Q and for Software Class N [[]]

]]

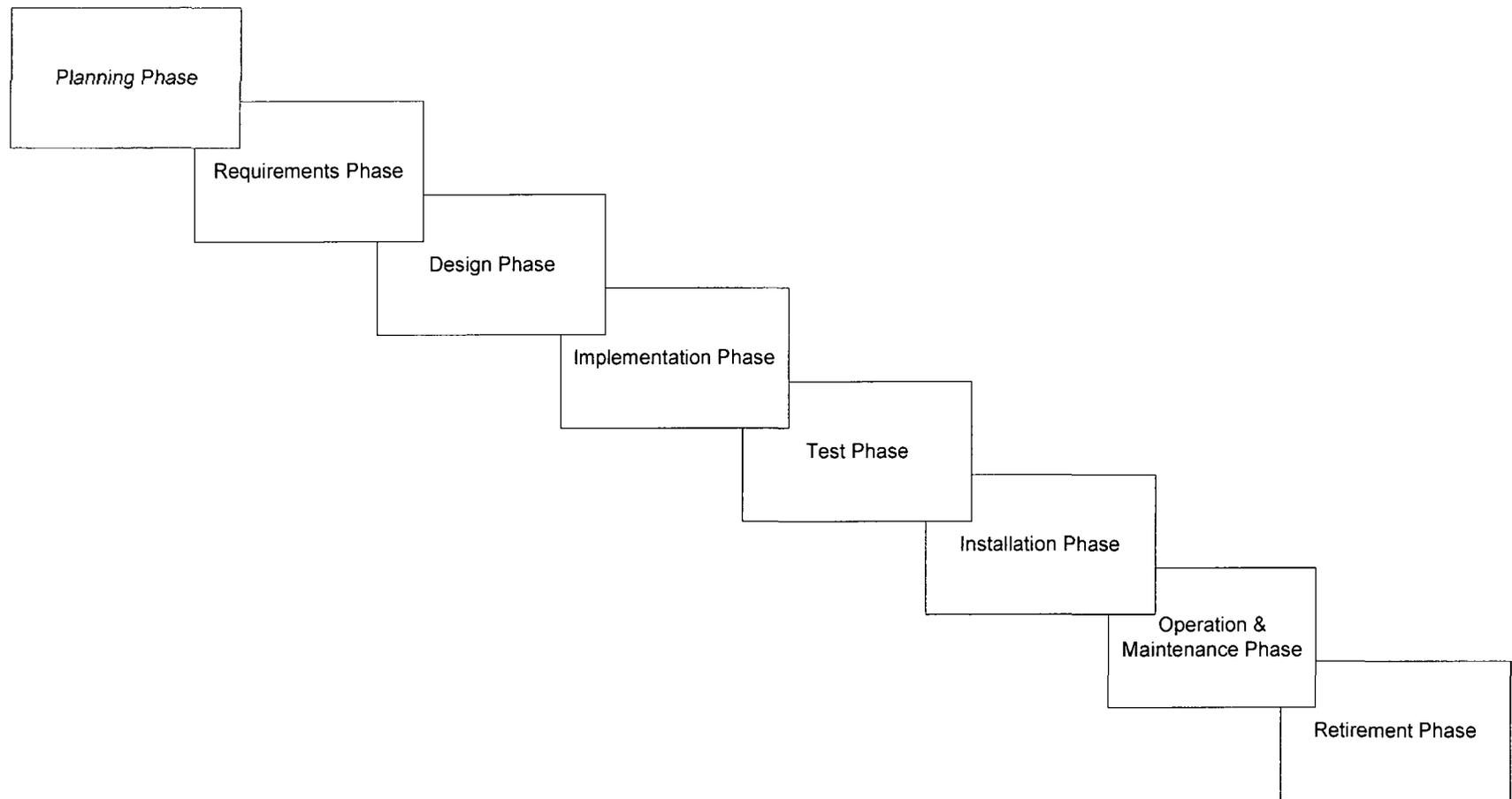


Figure 5-1. Software Life cycle Process Overview

[[

]]

[[

]]

[[

]]

[[

]]

[[

]]

[[

]]

[[

]]

[[

]]

[[

]]

[[

]]

[[

]]

II.

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

[[

o

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | | |
|--|--|--|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

6.0 SOFTWARE INTEGRATION PLAN

6.1.1 Overview

This Software Integration Plan (SIntP) consists of three major phases; integrating the various software modules together for form single programs, integrating the result of this the hardware and instrumentation and testing the resulting integrated product. During the first phase, the various object modules are combined to produce executable programs. These programs are then loaded in the second phase into test systems that are constructed to be as nearly identical as possible to the ultimate target systems, including computers, communications systems and instrumentation. The final phase consists of testing the results.

Software integration shall be performed during the software functional test.

6.1.2 Purpose

The purpose of this SMP is to:

- Describe the organization and responsibilities of individuals or groups involved in the SFT activities.
- Describe software functional test management, such as schedule, resources, security, risks and contingency planning, anomaly, problem reporting, and training needs.
- Provide a structure for software functional testing.
- Provide the requirements and guidelines necessary to prepare, execute, and document software functional tests.
- Define required software functional test documentation.
- Define measurements and metrics for error tracking and resolution, and assess the success or failure of the software integration and software test effort.

The approach to software integration and testing activities must be carried out in a deliberate and methodical manner.

For testing activities, deviations from this SMP shall be justified and approved by the RTPE. The justification and approval shall be prepared, reviewed, approved, and maintained in the software project DRF.

6.1.3 Software Integration

Software integration consists of three phases:

- Integrating the various software modules together to form single programs.
- Integrating programs with the hardware and instrumentation.
- Testing the resulting integrating product.

Interface analysis, data flow analyses, timing, and sizing analysis shall be performed as appropriate during software integration. The results of these analyses shall be documented in the

software functional test data sheet.

6.1.4 Organization and Management

The software functional test is performed by the software development organization. Section 6.2 describes the test personnel roles and responsibilities. SVVP [2.3(1), Section 7.0] describes the IVVT roles and responsibilities.

6.1.5 Management and Organizational Interfaces

The test results are reported by the RTE to the RTPE through the reports outlined in this document.

The software development and test team interface with the IVVT. The IVVT is responsible for performing the IV&V on the class Q Software Functional Test Report (SFTR).

6.1.6 Scheduling and Planning

The RTPE has overall responsibility for scheduling and planning test tasks and activities.

The schedule for software functional testing activities shall be integrated in the detailed work package as addressed in Section 5.0.

6.1.7 Resources

Resource management includes the determination of the required resources. Resources include the following elements:

1. Test facilities.
2. Test equipment and tools.
3. Qualified test engineers.
4. Any special needs for security, including cyber security.

6.1.8 Training

The IVVT Task Lead and the RTPE shall ensure that their staff is trained to support test activities.

6.1.9 Reviews

The progress of testing and issues related to testing shall be evaluated on a regular basis, such as during weekly review meetings. The progress report data from these meetings shall be used to track and update the project schedule. Special attention shall be paid to circumstances that indicate deficiencies in the testing process. Corrective actions shall be taken to improve the test process if a deficiency is identified using the CAR process as defined in Section 5.1.4.

6.2 Test Personnel Roles and Responsibilities

This section defines the test personnel roles and responsibilities.

6.2.1 The Responsible Technical Project Engineer (RTPE)

The RTPE, as part of the design team, has technical responsibility for the software functional test tasks.

6.2.2 Software Functional Test Engineer

The Software Functional Test Engineer, herein referred to as the, RTE is responsible for designing, executing, and documenting the test results in accordance with the SDD intra-system data communication protocol specification and this SMP.

6.2.3 Test Personnel Qualifications

The RTE shall be proficient in the targeted platform used, languages, software coding convention and guidelines, test techniques, and test tools.

6.3 Software Functional Test Guidelines

The following test guidelines include the key elements that are required for performing test activities:

1. Test preparation.
2. Test design.
3. Test execution.
4. Test summary.

This test guideline conforms to RG 1.170 [2.2.3(4)] and IEEE 829 [2.2.4(6)].

6.3.1 Test Preparation Guidelines

The purpose of test preparation is to assure that the required test activities can be properly carried out to ensure the software quality. This is accomplished by identification of resources required to support the development, execution, and the documentation of the test. The individual responsible for test preparation shall carry out the following tasks:

1. Define the scope of the test and identify the software items to be tested.
2. Design a detailed test schedule aligned with the project plan.
3. Specify test prerequisites.
4. Specify the test environment.
5. Identify equipment, documentation, tools, and instrumentation needed for the accomplishment of the test.
6. Adjust the integrated project schedule to account for equipment, documentation, tool, and instrumentation needs.
7. Assign qualified test designers(s) and tester(s).
8. Ensure the training needs are satisfied.
9. Start the test report.

6.3.2 Test Design Guidelines

The test designer shall perform the following tasks:

1. Specify the software features to be tested for each software item.
2. Specify the software features not to be tested and justify why they don't need to be tested (e.g., previously tested unmodified features and modifications have been demonstrated to not require re-test of these features).
3. Determine the test approach and specify the test techniques to be used.
4. Specify the test cases and acceptance criteria for each item.
5. Develop the test procedures and instructions.

Structural testing is a test methodology in which test steps are based on knowledge of the internal structure of the software module or a group of software modules. A structural test may execute all the statements or branches in the software module to check how the system is implemented. Methods to be used for structural testing include:

| | |
|-------------------|---|
| Branch testing | A testing technique to execute each outcome of each decision point in a computer program. |
| Path testing | A testing technique design to exercise every independent execution path through the computer program. |
| Statement testing | A testing technique design to execute each statement of a computer program. |

Functional testing is a test methodology that uses requirements external to a feature to derive test cases and test procedures. It verifies the end results at the feature I/O level, but does not check on how the feature is realized, nor does it assume that all statements related to the feature are executed. Methods to be used for functional testing include:

| | |
|--------------------------|--|
| Module interface testing | Testing performed to evaluate whether the values along the interface are correct as they relate to software modules that call them. |
| Interface testing | Testing performed to detect errors that may have been introduced into the system due to misinterpretation of the interface specification. |
| Regression testing | Selective re-testing of a software item to verify that modifications have not caused unintended effects and that the software item subject to the test still complies with its specified requirements. |
| Stress testing | Testing performed to evaluate a software item at or beyond the limits of its specified requirements. |

Reviews, in the form of design walkthroughs, are conducted during the test design process to

evaluate the adequacy of the selected test strategy and to assure that all test features are identified for software class Q.

6.3.3 Test Execution Guidelines

The purpose of test execution is to expose the software test item to conditions that may reveal potential implementation errors. Test execution includes the following tasks:

1. Obtain test items including relevant reference documentation.
2. Set the test environment.
3. Carefully observe the software and hardware during testing for both the expected and unexpected behaviors.
4. Confirm that completeness and test termination requirements are satisfied.
5. Document the test results while executing the test procedures.
6. Initiate the change control process per SCMP [2.3(1), Section 10.0] to resolve design errors encountered during the test.

6.3.4 Test Summary Guidelines

The purpose of the test summary is to evaluate the test. The test summary shall include:

1. Test activities.
2. Test results.
3. Requirement traceability.
4. Test issues and the associated resolutions.

The purpose of the requirement traceability is to demonstrate that every functional requirement, performance requirement, and interface requirement in a SDD and intra-systems data communication protocol specification have been verified by the test. The SQAP [2.3(1)] provides methods for performing traceability analysis.

[[

| | |
|--|--|
| | |
| | |

]]

7.0 SOFTWARE INSTALLATION PLAN

7.1 Introduction

The Software Installation Plan (SIP) describes the software installation process and activities which are performed during the Installation phase.

7.2 Purpose

The purpose of the SIP is to:

1. Define the installation phase activities.
2. Describe the installation procedures.
3. Describe the software installation management. This includes, but is not limited to schedule, resources, security, risks and contingency planning, anomaly and problem reporting, and training needs.
4. Provide the requirements and guidelines necessary to prepare, execute, and document software installation.

7.3 Scope

The scope of SIP is to address software installation strategy and techniques. The activities and procedures for the creation of documentation necessary to install software in the systems are discussed.

7.4 Organization, Management and Responsibilities

Organization activities are addressed in Section 3.0.

7.5 Installation Activities

The following sections define the activities to be performed during the Installation Phase of the software life cycle and include:

1. Software Installation Procedure.
2. Software Installation Reporting.
3. Installation Configuration Tables.
4. O&M Manuals.
5. Training Manuals.

7.5.1 Software Installation Procedure

A software installation procedure shall be produced for each software package. A combined procedure may be produced for multiple packages within a single system, but each system or logical group of systems should have its own installation procedure.

7.5.2 Software Installation Reporting

A software installation report shall be produced for each software installation procedure. A combined report may be produced for multiple packages within a single system, but each system or logical group of systems should have its own installation report.

For site installation, the Licensee shall produce and control software installation reports as part of the Licensee's change control, CM, and installation control processes.

7.5.3 Installation Configuration Tables

Where applicable, installation configuration tables shall be produced. The tables shall include all of the functional characteristics defined in the procedure section of the SIP to ensure that the software is correctly configured for the operating safety system.

For site installation, the Licensee shall produce and control installation configuration tables as part of the Licensee's change control, CM, and installation control processes.

7.5.4 Operations and Maintenance Manuals

The software O&M manuals shall be produced for each system or logical group of systems. Software O&M manuals shall include installation details necessary to enable the end user to install the software on the system.

7.5.5 Training Manuals

The software system training manuals for each system or logical group of systems shall be produced. The software system training manuals are based on design documents and the O&M manuals, and provide the basis for training the Licensee or end-user.

7.6 Procedures

The following sections define the procedural requirements for the software installation.

7.6.1 Software Installation Procedure

As a separate document or as a part of the O&M manual, the initial installation procedure for each individual software package and for each/or system or logical group of systems shall be defined for each plant software systems. The installation procedure shall include:

1. A description of the software installation procedure.
2. Software installation methods and procedures.
3. Criteria used to determine the success or failure of the installation effort.
4. A checklist or sequence of steps that can be used to confirm that the correct software is installed in the specific systems in accordance with the system design documents:
 - a. Affected functions are inoperable and in a safe condition according to the plant's technical specifications before proceeding with installation.

- b. The computer system is functional.
- c. The sensors and actuators are functional.
- d. All cards are present and installed in the correct slots.
- e. The communication system is correctly installed.
- f. The correct software versions are installed on the correct computers.
- g. Appropriate return-to-service testing has been successfully conducted before declaring the modified function operable.
- h. Installation configuration tables are complete.
- i. Environmental conditions, e.g., temperature, humidity, vibration, and rack space are considered and provided for.
- j. Identification of special tools, methods, or techniques used to accomplish the installation function is provided.
- k. Installation tools are to be qualified with a degree of rigor and level of detail appropriate to the safety significance of the software which are to be installed using these tools.
- l. Security provisions have been satisfied.
- m. Precautions to ensure personnel and plant safety have been identified.

The above list is a sample list of items that should be considered as a part of checklist.

7.7 Software Installation Report

An installation report for each package or system shall be produced upon the completion of the installation effort and shall report, as a minimum, a summary of installation activities that include:

- 1. Serial numbers or other identification for the hardware platform on which the software is installed.
- 2. Software revisions.
- 3. Circuit board revisions.
- 4. Any Cyclic Redundancy Code or checksum that may be displayed by the installed software.
- 5. Test results.
- 6. Anomalies discovered during installation.
- 7. Any associated data sheets generated during the installation.
- 8. Installation test summary.
- 9. Any user configurable parameter values.
- 10. Indication of Licensee approval and acceptance of the installation activities.
- 11. Completed checklist.

7.8 Installation Configuration Tables

When applicable, configuration tables shall be developed for each software system or logical group of systems. Each user configurable function shall be defined, along with each configurable mode, which includes the function, safety, and security of the overall application. The configuration tables shall include the following items:

1. Software configuration tables shall include all information necessary for the correct operation of the system and its associated plant functions. This includes any vendor default settings used to test and accept the initial configuration.
2. Installation configuration tables shall be consistent with the software specifications, as described in the SRS, SDD, software code, and software build documents.
3. Software configuration tables shall contain system specific data.
4. Class Q software shall be required to provide traceability for each installed program element backward to the integrated software elements that created that installed program element.

7.9 Training Manuals

Training manuals are described in Subsection 9.6.1.

7.10 Installation Methods

Installation methods, tools, software, and hardware used to perform software installation shall be defined in each software installation manual as required for each software package.

7.11 Software Archive Retrieval

The software package shall be placed under CM as required by the SCMP [2.3(1) Section 10.0]. Plant-specific methods of archival and retrieval are the responsibility of the Licensee and are beyond the scope of this SMP. Where applicable, specific backup and recovery procedures shall be included in the maintenance section of the O&M manual.

7.12 Software Installation Test

A software installation test and procedure shall be developed as a separate test procedure or as part of the installation procedures for each software package to be installed.

7.13 Installation Documentation and Problem Reporting

The problem or issues encountered during installation process shall be reported in an installation report. The SQAP [2.3(1)] defines a process for problem reporting and corrective action.

7.14 Verification and Validation Methods

The installation phase outputs shall be verified in accordance with the SVVP, [2.3(1) Section 7.0].

7.15 Measurements and Metrics

Measurement and metrics shall be developed in accordance with the SQAP [2.3(1)].

8.0 SOFTWARE OPERATIONS AND MAINTENANCE PLAN

8.1 Introduction

The Software Operation and Maintenance Plan (SOMP) defines the software process and activities used to operate and maintain the software product during plant operation.

8.1.1 Purpose

This SMP defines requirements, methods, and considerations for developing the system O&M manual. This SMP also addresses maintenance procedures and activities to enhance, modify, and maintain software once the software is installed in the plant.

8.1.2 Scope

The scope of this SMP is to address the activities for the software product for the Operations and Maintenance Phase. The guidance for these activities is generally provided to the Licensee or end-user through the O&M Manual. The SQA requirements are addressed in the SQAP [2.3(1)].

8.2 Organization, Management and Responsibilities

Organization activities are addressed in Section 3.0. Management activities are addressed in Section 5.0. Responsibilities are addressed in Section 3.4.

8.3 Activities

The plans, procedures, processes, and activities for software corrections and for software enhancements in the O&M phase are the same as those used in the Planning, Requirements, Design, Implementation, Test, and Installation Phases. The following sections define the activities to be performed during the O&M phase of the software life cycle:

8.3.1 Operation Phase Activities

The O&M manual, developed in the Installation phase, defines procedures for specific recommended activities to be performed during the operation of the system in which the software is installed. The O&M manual may be used to develop plant specific procedures, which include:

1. Monitoring the software to detect security breaches, including penetration or attempted penetration of the system.
2. Measuring, recording, evaluating root cause, analyzing, and reporting system errors and error rates.
3. Surveillance procedures for ensuring that:
 - a. The system is operating correctly and is calibrated.
 - b. The software state is consistent with the plant-operating mode.
4. The system is ready and able to perform its safety-related function, for all Class Q software

5. The system is ready and able to control and monitor plant operation, for Class N software
6. Developing backup and restore procedures for configuration, data, and code.
7. Developing calibration procedures.
8. Developing maintenance procedures.

8.3.2 Maintenance Phase Activities

Maintenance or design engineering change control is the process used to control, authorize, and implement changes to engineering controlled CIs. The same software plans, processes, and procedures used to correct errors in the software during the initial life cycle phases shall be used to make corrections and enhancements to the software after the system is installed in the plant. The Licensee may continue software life cycle activities using a set of plans, procedures, and processes based on the original software project life cycle development plans. The Licensee shall establish appropriate contractual arrangements with the designer and software developer before system turn over. The Licensee's plans, processes, and procedures shall provide any additional required installation, commissioning, testing, change control, and CM procedures.

8.4 Procedures

The following sections define the procedural requirements for each O&M phase activity.

8.4.1 Operation Phase Procedures

Generic operating procedures of the software product shall be defined in the O&M manuals. Plant specific procedures, which are the responsibility of the Licensee, shall be developed using the generic procedures supplied by the operation, maintenance, and training manuals. Both GEEN and the Licensee shall have a 10 CFR 50 Appendix B, QA program and Part 21 reporting system in place to notify each other promptly of any Class Q software nonconformance. Similar mechanisms shall be in place with the Class N software vendors to ensure prompt notification of detected errors and their resolutions. The Licensee shall establish a single identified functional POC within the Licensee's organization for any system error notification. The operations procedures shall include:

- A description of the responsibilities and authority of the operators.
- A description of the security requirements for operating the software system.
- Identification of the controls needed for operational activities to prevent unauthorized changes to hardware, software, and system parameters.
- The monitoring activities needed to detect penetration or attempted penetration of the system, and contingency plans needed to ensure appropriate response to penetration.

8.4.2 Maintenance Phase Procedures

Maintenance is the process of maintaining and monitoring software performance. The maintenance procedures shall include:

- A description of the method used for software risk management during maintenance, with particular attention to risks that have the potential for compromising safety.
- A description of the methods used to prevent contamination viruses, trojan horses, or other nefarious additions.
- The required security level for each maintenance task.
- Measures to minimize the potential for introducing unauthorized changes during repair, testing, and calibration.

The maintenance procedures shall follow the entire software life cycle, from planning through re-installation.

8.5 Methods and Tools

Methods and Tools to perform software O&M shall be defined in the O&M manual for each software package/system or logical group of systems. The O&M manual shall include a description of the configuration control required to maintain the delivered software. The O&M manual shall list and describe the software, hardware, and associated documentation required to maintain the delivered software. Maintenance tools shall be qualified to the level associated with the safety significance of the software.

[[

]] The operation section of O&M Manual shall include a description of the actions available to the operator/user as listed below:

1. The operating modes.
2. Error messages, including description and error recovery methods.
3. Backup and recovery procedures.
4. Operator actions shall be specified in terms of inputs supplied by the operator or system.
5. Actions initiated by the operator.
6. Responses to the operator or system.

The purpose and operation of each function shall be described, including interfaces with other systems. The O&M manual shall describe methods, techniques, tools, software, hardware, and associated documentation required to operate the software.

The O&M Manual shall describe the operational environment within which the software shall operate, including:

1. Precautions.
2. Limitations.
3. Personnel or plant hazards.
4. Security vulnerabilities.

5. Variables in the physical environment that the software must monitor and control.
6. User interfaces. User interfaces shall be described fully for each category of operator or user.

The operations section of the O&M manual shall be consistent with the system operations, system requirements, the system design, (i.e. SRS, SDS or SDD,) documented descriptions, and known properties of the operational environment within which the software shall operate. Individual user instructions shall not contradict other instructions. Uniform and consistent terminology, notation, and definitions shall be used throughout the manuals. Vendor supplied manuals shall adhere to the same requirements.

The maintenance manual section shall include the following:

1. Precautions.
2. Limitations.
3. Personnel or plant maintenance hazards.
4. Security vulnerabilities.
5. Trouble shooting and reporting procedures and methods.

Configuration management and change control procedure shall be described in or referenced by the maintenance section of the O&M manual. Procedures shall exist to:

1. Verify that changes have been implemented correctly, the changes and a sufficient test overlap have been defined and performed, and that no faults have been introduced in the software by the changes.
2. Ensure that software is functioning properly after the maintenance.
3. Upgrade field procedures. Field upgrade procedures shall be described, including:
 - a. Installation procedures.
 - b. Installation test procedures.
 - c. Installation test checklists.
 - d. Installation test data sheets

8.5.1 Software Operations Maintenance Manuals

The Software O&M Manual shall be developed in accordance with the requirements outlined in the Software O&M Plan and the HFE/MMIS IP and may be incorporated into the System O&M Manual.

The software operation manual shall, at a minimum, include the following:

1. Information necessary for all operating modes. This includes normal operation, off normal operation, and emergency operation.
2. Start-up and shut-down of the software product. This includes error recovery and backup.

3. Error messages. Error messages shall be listed together with their meaning and corrective action by the operator.
4. Description of the operational environment within which the software shall operate, including precautions and limitations that must be observed during operations to avoid exposing personnel or the plant to hazards or security vulnerabilities.
5. Description of each user interfaces for each category of user, including operators, shift supervisors and, nuclear engineers.

The software maintenance manual shall, at a minimum, include the following:

1. Description of the procedures to be followed when operational software must be changed.
2. Identification of the precautions and limitations that must be observed during maintenance to avoid exposing personnel or the plant to hazards or security vulnerabilities.
3. Change control process. The CM shall be described or referenced.
4. Software Installation Procedure, including regression test steps to be executed to confirm the revised/enhanced software is correctly installed, and that no faults have been introduced in the revised/enhanced software.
5. Methods used to restore older versions of software and methods used to back up software.
6. Methods to troubleshoot and diagnose both the system and its inter-connects to the MMIS.

8.5.2 Verification and Validation Methods

The O&M phase outputs shall be verified in accordance with the SVVP, [2.3(1) Section 7.0].

8.6 Measurement and Metrics

Measurements and metrics shall be developed in accordance with the SQAP [2.3(1)].

9.0 SOFTWARE TRAINING PLAN

9.1 Introduction

This Software Training Plan (STrngP) describes the software training activities to be carried out before and during the operation of software products for the plant. Software training is performed prior to delivery of the software (system startup and post turn over) and during the O&M phase of the software life cycle. The STrngP addresses the management, implementation and resource characteristics as addressed in BTP-14 [2.2.1]. The STrngP also adheres to the HFE requirements for training as outlined in the HFE/MMIS IP [2.1(1)].

9.1.1 Purpose

The purpose of the STrngP is to define:

- The requirements and methods to use while developing the training manual.
- The training needs of appropriate plant staff, including operators, I&C engineers, and technicians.
- A general description of the training facilities.
- The organization supporting the training effort, including interfaces and responsibilities.

9.1.2 Scope

The scope of this SMP is to address the training requirements and documentation for each system or logical group of systems needed to ensure proper operation and use of the software within the overall system. These training requirements include proper usage, (i.e., personal safety, system security) of the equipment for the users, operators, maintenance personnel, and management personnel. This SMP describes the approach for identifying training requirements for use in developing the related training documents.

9.2 ESBWR-Training Organization

This section provides a description of the ESBWR Training organization supporting the software product training effort, as well as organizational interfaces and responsibilities. Figure 3-1 shows the relationship of the training organization which reports to GEEN Nuclear Services. The organizational responsibilities are identified in Section 3.4. The TSL is a functional position responsible for assignment of personnel to support training for the software products. The TSL is responsible for ensuring that the training requirements are accomplished. The training requirements are established based on Licensee needs to generate and maintain the software products. The TSL augments the training staff to support the required training based on the Licensee needs.

9.2.1 Responsibilities and Qualification

The TSL has overall responsibility for the trainer qualification process. Qualified personnel are

selected for Trainer positions based on work related experience and knowledge in the operation of Nuclear Power Plant and I&C Systems, as detailed in the individual's resume.

9.3 Training Activities

This section defines the required training activities training activities include:

1. Development and maintenance of training plans.
2. Development and review of the training manual.
3. Development of training courses.
4. Development of training.

Plant specific training procedures, as defined by IEEE 1074, are post development activities and are the responsibility of the Licensee.

The training manual should address the following activities:

1. Startup.
2. Shutdown.
3. Installation.
4. Backup.
5. Restoration.
6. Configuration.
7. Calibration.
8. Troubleshooting.
9. Replacing failed modules.
10. Plant modes, including alarm and indicator responses.
11. Training assessment.
12. Operating specific scenarios.
13. Recommended surveillance testing.
14. Security, including Cyber Security.

9.3.1 Software Training Manual Program

Software training manuals shall include the following requirements:

1. A full description of actions available to the operator and to the technician for all operating modes, including error recovery.
2. A description of operator actions. Description of operator actions shall be specified in terms of inputs supplied by users and equipment, actions initiated by the operation, and responses to the user.

3. A description of the maintenance environment, including precautions and limitations that must be observed during maintenance to avoid incorrectly configuring, damaging, or otherwise defeating the system's functionality and thus exposing the plant to hazards.
4. A description of the operational environment within which the software shall operate, including precautions and limitations that must be observed during operations to avoid exposing personnel or the plant to hazards.
5. A full description of variables in the physical environment that the software must monitor and control. User interfaces should be fully described for each category of user.

The Training Manual shall be prepared in accordance with the requirements specified in the STRngP and the HFE/MMIS IP. The training manual shall be completed and accepted by the Licensee prior to the start of the training sessions. The timing of the Licensee's acceptance shall be as specified in the contract. Software training manuals are described in Section 9.6.1 of this document.

9.4 Training Program

A comprehensive training program with a comprehensive set of established training modules or programs shall be developed for the software products. Training is provided for the following generic types of system users:

- Plant Operations.
- Maintenance.
- System Administrator.
- General Purpose User.
- Engineering.

ESBWR HFE Training Development Implementation Plan [2.3(5)], describes the processes, methods, and criteria for the development of the ESBWR training program, including the programs for operations, maintenance, and support of the software products.

9.5 Methods and Tools

Methods and tools used to perform software training shall be defined in each manual as required for each software system or logical group of software systems. The responsible trainer shall determine each training course's content and methods. The TSL shall approve all course content and methods.

9.6 Training Facilities

Effective training requires effective training facilities to fulfill the training objectives. When preparing a training course, the trainer must determine the type of training facility that shall provide the most effective nuclear training. Examples of training facilities that may be utilized are:

- Dedicated classroom space such as use of conference rooms when available.

- Instructor led Classroom software Lab facilities.
- Self Study Computer lab facilities.
- A remote training access tool, such as, presentation tools which allow training at a remote training workstation.
- Control Room Simulator.

9.7 Metrics

The use of metrics provides a basis for determining the effectiveness of the training program. Metrics selected during the development of the training program may be a combination of a number of tools based on the nature of the training program being offered. For example, a training course that provides an overview during a one day session would utilize a different set of metrics than a four week course that utilizes extensive use of simulation training tools. The program should allow for quizzes or practical exams based on course objectives relevant to the task responsibilities. The training program may also allow self-study for certain aspects of the training.

Examples of training tool metrics that may be used are:

- Instructor Assessment. The instructor queries trainees during class session and grades daily performance.
- The Certification exams.
- Computer software lab tests.
- Student performance during plant scenarios in a training simulator.

The test results or training results obtained at the end of the training activities shall be measured, recorded, analyzed, and reported.

[[

| | |
|--|--|
| | |
| | |
| | |
| | |
| | |

]]

APPENDIX A - Acronyms and Abbreviations

The following acronyms and abbreviations are used throughout this SMP.

| Acronym | Meaning |
|---------|--|
| ADS | Automatic Depressurization System |
| ANSI | American National Standard Institute |
| AOF | Allocation of Function |
| ASQ | American Society for quality |
| ASL | Approved Suppliers List |
| ASME | American Society of Mechanical Engineers |
| ATWS | Anticipated Transients without Scram |
| BD | Build Description |
| BFM | Business Finance Manager |
| BR | Baseline Review |
| BRD | Build Release Description |
| BRR | Baseline Review Record |
| BRT | Baseline Review Team |
| BTP | Branch Technical Position |
| CAQ | Condition Adverse to Quality |
| CAR | Corrective Action Request |
| CCB | Change Control Board |
| CEO | Chief Executive Officer |
| CI | Configuration Item |
| CM | Configuration Management |
| CMM | Configuration Management Manager |
| CMS | Configuration Management System |
| CMU | Configuration Management Unit |
| COL | Combined Operating License |
| COTS | Commercial-Off-The-Shelf |

| Acronym | Meaning |
|----------------|---|
| CTS | Commitment Tracking System |
| DCD | Design Control Document |
| DLD | Detailed Logic Diagram |
| DRF | Design Record File |
| ECA | Engineering Change Authorization |
| ECN | Engineering Change Notice |
| eDRF | Electronic Design Record File |
| EM | Engineering Manager |
| EOP | Engineering Operating Procedure |
| EPRI | Electrical Power Research Institute |
| ERM | Engineering Review Memorandum |
| ESBWR | Economic Simplified Boiling Water Reactor |
| FAPCS | Fuel and Auxiliary Pools Cooling System |
| FAT | Factory Acceptance Test |
| FDDR | Field Deviation Disposition Request |
| FDI | Field Disposition Instruction |
| FMEA | Failure Modes and Effects Analysis |
| FRA | Functional Requirements Analysis |
| FX | Function |
| GE | General Electric Company |
| GEEN | GE Energy Nuclear |
| HFE | Human Factors Engineering |
| HICB | Instrumentation and Control Branch |
| HSI | Human System Interface |
| HSS | Hardware/Software Specification |
| I&C | Instrumentation and Control |
| I&C EEM | Instrumentation and Controls Electrical Engineering Manager |
| I&C ESE | Electrical Systems Engineer |

| Acronym | Meaning |
|---------|--|
| IDT | Implementation Design Team |
| IEEE | Institute of Electrical and Electronic Engineers |
| IMS | Information Management System |
| IP | Implementation Plan |
| ISO | International Standards Organization |
| IV&V | Independent Verification and Validation |
| IVVT | Independent Verification and Validation Team |
| LTR | Licensing Topical Report |
| MCR | Main Control Room |
| [[|]] |
| MMIS | Man Machine Interface System |
| N/A | Not Applicable |
| NPP | New Plant Project |
| NQA | Nuclear Quality Assurance |
| O&M | Operation and Maintenance |
| P&ID | Piping & Instrumentation Diagram |
| P&P | Policies and Procedure |
| PDMS | Product Data Management System |
| PDS | Previously Developed Software |
| PE | Project Engineer |
| PHA | Preliminary Hazards Analysis |
| PM | Project Manager |
| PMT | Project Management Team |
| PO | Purchase Order |
| POC | Point of Contact |
| PQC | Product Quality Certification |
| PRA | Probabilistic Risk Assessment |
| PWP | Project Work Plan |

| Acronym | Meaning |
|----------------|--|
| QA | Quality Assurance |
| QCE | Quality Control Engineer |
| RCCE | Responsible Configuration Control Engineer |
| RE | Responsible Engineer |
| RG | Regulatory Guide |
| RMCN | Review Memorandum Change Notice |
| RTA | Requirements Traceability Analysis |
| RTE | Responsible Test Engineer |
| RTM | Requirements Traceability Matrix |
| RTPE | Responsible Technical Project Engineer |
| RV | Responsible Verifier |
| SAE | Simulation Assisted Engineering |
| SAT | Site Acceptance Test |
| SATT | Site Acceptance Test Team |
| SCM | Software Configuration Management |
| SCMP | Software Configuration Management Plan |
| SDD | Software Design Description |
| SDP | System Development Plan |
| SDS | System Design Specification |
| [[|]] |
| SFRA | System Functional Requirements Analysis |
| SFT | Software Function Test |
| SFTR | Software Functional Test Report |
| SIntP | Software Integration Plan |
| SIP | Software Installation Plan |
| SITT | System Integrated Test Team |
| SMP | Software Management Plan |
| SOMP | Software Operations and Maintenance Plan |

| Acronym | Meaning |
|----------------|---|
| SPE | Software Project Engineering |
| SQA | Software Quality Assurance |
| SQAP | Software Quality Assurance Plan |
| SQAT | Software Quality Assurance Team |
| SRP | Standard Review Plan |
| SRS | Software Requirements Specification |
| SSA | Software Safety Analysis |
| SSP | Software Safety Plan |
| SST | Software Safety Team |
| STmgP | Software Training Plan |
| SVVP | Software Validation and Verification Plan |
| SyAT | System Acceptance Testing |
| SyRS | System Requirement Specification |
| TA | Task Analysis |
| TPE | Technical Project Engineer |
| TSL | Training Services Lead |
| USNRC | United States Nuclear Regulatory Commission |
| V&V | Verification and Validation |
| VTE | Validation Test Engineer |

APPENDIX B - Definitions

| Term | Definition |
|------------------------------|--|
| Acceptance Criteria | The criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity [IEEE 610.12]. |
| Acceptance Testing | Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system [IEEE 610.12]. |
| Algorithm | A finite set of well-defined rules for the solution of a problem in a finite number of steps [IEEE 610.12]. |
| Anomaly | Anything observed in the documentation or operation of software that deviates from expectations based on previously verified software products or reference documents [IEEE 610.12]. |
| Application software | Software designed to fulfill specific needs of a user [IEEE 610.12]. |
| Application Software Package | A collection of software modules brought together to form a single software application, e.g., an instrument (see also System Software Package and Package). |
| Assembly code | Computer instructions and data definitions expressed in a form that can be recognized and processed by an assembler. |
| Baseline | Items that have been formally reviewed and agreed upon, that thereafter serve as the basis for further development, and that can be changed only through formal change control procedures [IEEE 610.12]. |
| Baseline Review | A formal review, conducted at the end of each process step of the software engineering design process, and requested by the Design Team's responsible TPE. The baseline review process is under the control of Software Project Engineering (SPE). The Baseline Review Team (appointed by the BRT Task Lead engineer) performs the review. These reviews are intended to confirm adherence to the project SMP and SCMP. The Baseline Reviews are performed and documented in accordance with the Software Configuration Management Plan, the Software Quality Assurance Plan, and the Software Verification and Validation Plan. |
| Branch testing | Testing designed to execute each outcome of each decision point in a computer program [IEEE 610.12]. |
| Build | An operational version of a system or component that incorporates a specified sub set of the capabilities that the final product will provide [IEEE 610.12]. |

| Term | Definition |
|----------------------------|--|
| Certification | A written guarantee that a system or component complies with its specified requirements and is acceptable for operational use [IEEE 610.12]. |
| Code | In software engineering, computer instructions and data definitions expressed in a programming language or in a form output by an assembler, compiler or other translator [IEEE 610.12]. |
| Code review | A meeting at which software code is presented to project personnel, managers, users, customers, or other interested parties for comment or approval [IEEE 610.12]. |
| Coding | In software engineering, the process of expressing a computer program in a programming language [IEEE 610.12]. |
| Commitment Tracking System | System used to manage the Conditions Adverse to Quality (CAQs). A Corrective Action Request (CAR) is used to document a CAQ, or an opportunity for process/product improvement, provide for timely evaluation, and record objective evidence of actions taken. EOP 75-3.00, Self-Assessment, Corrective Action and Audits [2.3(2x)] specifies the responsibilities for actions to promptly identify, record and correct, as appropriate, CAQs, and to assure that these conditions do not affect the quality of a product or service. |
| Component | One of the parts that make up a system. A component may be hardware or software and may be subdivided into other components [IEEE 610.12]. |
| Computer language | A language designed to enable humans to communicate with computers [IEEE 610.12]. |
| Configuration control | An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification [IEEE 610.12]. |
| Configuration Item | An aggregation of hardware, software, design documents or procedures that is designated for configuration management and treated as a single entity in the configuration management process [IEEE 610.12]. |
| Design Documentation | Design Documentation is information recorded about a specific life cycle activity. Documentation includes software life-cycle design outputs and software life cycle process documentation. A document may be in written or electronic format, and may contain text, illustrations, tables, computer files, program listings, binary images, and other forms of expression. A document for an activity may be packaged with documents for other activities, or documents for non-software life cycle activities. A document for an activity may be divided into several individual entities. |

| Term | Definition |
|-------------------------------------|--|
| Design output | Documents, such as drawings and specifications, that define technical requirements of structures, systems, and components. For software, design outputs include the products of the development process that describe the end product that will be installed a nuclear power plant. The design outputs of a software development process include SRS, SDD, hardware and software architecture designs, code listings, system build documents, installation configuration tables, O&M manuals, and training manuals, reviews, and test records. |
| Design phase | The <i>phase</i> in the software life cycle during which the designs for architecture, software components, interfaces, and data are created, documented, and verified to satisfy requirements [IEEE 610.12]. |
| Design Record File | A formal controlled information record under GEEN procedures for in-progress and completed engineering work which is retained and from which work can be retrieved. |
| Design Reviews | Formal, design adequacy evaluations which are performed by knowledgeable persons other than those directly responsible and accountable for the design in accordance with EOP 40-7.00. Design reviews are used to verify that product designs meet functional, contractual, safety, regulatory, industry codes and standards, and company requirements. |
| Deviation | A departure from a specified requirement. |
| Documentation | A collection of documents on a given subject [IEEE 610.12]. |
| Error | An incorrect step, process, or data definition [IEEE 610.12]. |
| Failure Mode and Effects Analysis | A tabular method of providing traceability from the modes by which a system may fail and the effect of that failure on the ability of the system to perform its function, or the ability of a collection of systems to recover from the failure. |
| Fault Tree | A pictorial method of providing traceability from the modes by which a system may fail and the effect of that failure on the ability of the system to perform its function, or the ability of a collection of systems to recover from the failure. |
| Field Deviation Disposition Request | Field Deviation Disposition Request (FDDR) is used for documenting and disposition of the technical position for a deviation required in the field in supplied hardware, software, or services (see EOP 55-3.00). |
| Firmware | The combination of a hardware device and computer instructions and data that reside as read-only software on that device [IEEE 610.12]. |

| Term | Definition |
|--|--|
| Functional Testing | A system/software test methodology that is derived from external specifications and requirements of the system. Such testing ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions [IEEE 610.12]. Methods for functional testing include random testing and testing at boundary values. It verifies the end results at the system level, but does not check the implementation techniques, nor does it assume that all statements in the program are executed. |
| Implementation Phase | The <i>phase</i> in the software life cycle during which a software product is created from design documentation and debugged [IEEE 610.12]. |
| Independent Verification and Validation (IV&V) | Verification and Validation performed by an Organization that is technically managerially, and financially independent of the Organization [IEEE 610.12] and RG 1.168 Section C3 [2.2.3(1)]. |
| Installation Phase | The <i>phase</i> in the software life cycle during which the software product is installed into its operational environment and tested to ensure that it performs as intended [IEEE 610.12]. |
| Instrument | A hardware device used for analytical or control functions and usually containing on or more embedded microprocessor. |
| Integration Testing | Testing in which software elements, hardware elements, or both are combined and tested to evaluate the interaction between them [IEEE 610.12]. |
| Interface | A shared boundary across which information is passed [IEEE 610.12]. |
| Metric | A quantitative measure of the degree to which a system, component, or process possesses a given attribute [IEEE 610.12]. |
| Module | A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to, or output from, and assembler, compiler, linkage editor, or executive routine [IEEE 610.12]. |
| Operations and Maintenance Phase | The <i>phase</i> in the software life cycle during which the software product is functioning in its operational environment, monitored for satisfactory performance and modified as necessary to correct problems or to respond to changing requirements [IEEE 610.12]. |
| Package | A separately compliable software component consisting of related data types, data objects and sub-programs [IEEE 610.12]. |
| Path Testing | Testing designed to execute all or selected paths through a computer program [IEEE 610.12]. |
| Planning Phase | The initial <i>phase</i> of a software development project, in which project scope, purpose, strategy, schedule and milestones are established and user needs through documentation (for example, system definition documentation and procedures) are described and evaluated. |

| Term | Definition |
|---|--|
| Procedure | A course of action to be taken to perform a given task [IEEE 610.12]. |
| Process | A sequence of steps performed for a given purpose, e.g., the software development process [IEEE 610.12]. |
| Project <i>Management</i> Plan | A document that describes the technical and management approach to be followed for a project. The plan typically describes the work to be done, the resources required, the methods to be used, the procedures to be followed, the schedules to be met, and the way that the project will be organized [IEEE 610.12]. |
| Software Class N2 | Nonsafety-related system software whose failure cannot adversely affect a safety-related function. |
| Software Class N3 | <p>Nonsafety-related systems software whose failure could challenge safety systems as defined below:</p> <ul style="list-style-type: none"> a) Software whose inadvertent response to stimuli, failure to respond when required, response out-of-sequence could directly result in an accident or transient as defined in the DCD, chapter 15 [2.1(4)]. b) Software that is intended to mitigate the result of an accident. c) Software that is intended to recover from the result of an accident. |
| Software Class Q | System software which performs functions that are classified per EOP 65-2.10 as safety-related. |
| Regression Testing | Selective re-testing of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements [IEEE 610.12]. |
| Requirement | <p>A condition or capability that must be met or possessed by a system or system component to satisfy a contract standard specification or other formally imposed documents [IEEE 610.12].</p> <p>In specifying requirements, the word shall is used to indicate mandatory requirements and from which no deviation is permitted ('shall' and 'required to' are equivalent in meaning).</p> <p>Requirements are not specified with the word should. Instead, should is used to indicate that a recommended course of action and is particularly suitable, without mentioning or excluding other courses of action; Also, a certain course of action is preferred but not necessarily required; Also, that (in the negative form) a certain course of action is not prohibited ('should' and 'recommended' are equivalent in meaning).</p> |
| Requirements Phase | The <i>phase</i> in the software life cycle during which the requirements for a software product are defined and documented [IEEE 610.12]. |
| Requirements <i>Traceability</i> Analysis | The process of studying user needs to arrive at a definition of system, hardware, or software requirements [IEEE 610.12]. |

| Term | Definition |
|--|---|
| Responsible Configuration Control Engineer | The person assigned responsibility for the configuration management of the I&C software products. |
| Responsible Engineer | The person responsible for a given technical item, e.g., the design and development of the documentation. |
| Responsible Technical Project Engineer | The person with overall technical responsibility for ensuring that the hardware and software design of a software product meets the specified requirements. |
| Responsible Verifier | The Responsible Verifier(s) is an individual who has the independence described in EOP 42-6.00 for verifications, or in EOP 42-6.10 for deferred verifications of design process and the accompanying documents. |
| Retirement | Permanent removal of a system or component from its operational environment [IEEE 610.12]. |
| Simulation | A model that behaves or operates like a given system when provided a set of controlled inputs [IEEE 610.12]. |
| Software Development Process | The process by which user needs are translated into a software product. The process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code, and sometimes, installing and checking out the software for operational use [IEEE 610.12]. |
| Software Feature | A distinguishing characteristic of a software item, such as, performance, portability, or functionality. |
| Software Item | Source code, object code, job control code, control data, or a collection of these items [IEEE 610.12]. |
| Software Life cycle | The period of time that begins when a software product is conceived and ends when the software is no longer available for use [IEEE 610.12]. |
| Software Life cycle Phase | The division of the software life cycle into discrete logical units. The I&C software life cycle is divided into nine <i>phases</i> , namely, Planning, Requirements, Design, Implementation, Integration, Validation, Installation, and Operation & Maintenance, and retirement. |
| Software Module | See Module. |
| Software Package | See Package. |
| Software Unit | See Module. |
| Source Code | Computer instructions and data definitions expressed in a human readable form suitable for input to an assembler, compiler, or other translator. |

| Term | Definition |
|----------------------------|---|
| Statement testing | Testing designed to execute each statement or a computer program [IEEE 610.12]. |
| Stress testing | Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements [IEEE 610.12]. |
| Supplemental Document | Controlled documents that are referenced or used in conjunction with this SMP. These are the enabling documents that either augment or enable the performance of the activities stated in this SMP. |
| Support software | Software that aids in the development or maintenance of other software; for example, compilers, loaders, and other utilities [IEEE 610.12]. |
| Supporting Document | Controlled documents used in the production of this SMP. These documents form the design basis for the activities stated in this SMP. |
| System Testing | Testing conducted on a complete, integrated system to evaluate the systems compliance with its specified requirements [IEEE 610.12]. |
| Technical Project Engineer | The person with overall technical responsibility for ensuring that the hardware and software design of a software product meets the specified requirements. |
| Test case | A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement [IEEE 610.12]. |
| Test Item | A software item that is an object of testing [IEEE 610.12]. |
| Test Log | A chronological record of all relevant details about the execution of a test [IEEE 610.12]. |
| Test Objective | An identified set of software features to be measured under specified conditions by comparing actual behavior with the required behavior described in the software documentation [IEEE 610.12]. |
| Test Phase | The <i>phase</i> in the software life cycle during which the components of a software product are integrated with the hardware and evaluated to determine whether or not performance requirements have been satisfied [IEEE 610.12]. |
| Test Plan | A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do such task, and any risks requiring contingency planning [IEEE 610.12]. |
| Traceability Matrix | A matrix that records the relationship between two or more product specifications (i.e., design documentation) of the development process (e.g., a matrix that records the relationship between the requirements and the design of a given software component) [IEEE 610.12]. |
| Unit Testing | Testing of individual hardware or software units or groups of related units [IEEE 610.12]. |

| Term | Definition |
|-----------------------------------|---|
| User interface | An interface that enables information to be passed between a human user and hardware or software components of a computer system [IEEE 610.12]. |
| Verification and Validation (V&V) | The design verification activities performed in accordance with GEEN EOPs 40-7.00 (Design Reviews) or 42-6.00 (Independent Design Verification) based on 10CFR50 Appendix B [2.2.2(1)] or equivalent to ensure the quality of the design process and the associated documents produced. For Class Q software products, the verification and validation activities are performed by the SPE in accordance with the design process (SVVP) to ensure the quality of the associated documents produced. |

APPENDIX D - Software Functional Test Metrics Sheet (example)

SOFTWARE FUNCTION TEST METRICS SHEET DRF#

Page: 1 of 1

Application Software Package: Revision:

Subsystem: Software Class:

| <i>Total Errors</i> | <i>Error Type</i> | <i>SFT Section</i> |
|---------------------|---|--------------------|
| | Data reference errors – errors that occur when data items are referenced improperly, | |
| | Data declaration errors – errors resulting from conflicts between intended and actual usage, | |
| | Computation errors – errors resulting from improper analysis or computational precision, | |
| | Comparison errors – errors resulting from improper or imprecise condition expressions, | |
| | Control flow errors – errors resulting from incorrect branching targets, | |
| | Interface errors – errors resulting from improper passage of data between software modules, | |
| | Input/Output errors – errors resulting from incorrect data formats or invalid interface specifications. | |
| | Hardware errors, and | |
| | Hardware/Software interaction errors, | |
| | Task Interaction errors, | |
| | Other errors | |
| Totals | | |

Notes

APPENDIX E - Software Validation Test Metrics Sheet (example)

SOFTWARE VALIDATION TEST METRICS SHEET DRF#

Page: 1 of 1

Application Software Package: Revision:

Subsystem: Software Class:

| Total Errors | | Error Type | SFT |
|--------------|-------|--|---------|
| Major | Minor | | SECTION |
| 0 | 0 | Data reference errors - errors that occur when data items are referenced improperly, | |
| 0 | 0 | Data declaration errors - errors resulting from conflicts between intended and actual usage, | |
| 0 | 0 | Computation errors - errors resulting from improper analysis or computational precision, | |
| 0 | 0 | Comparison errors - errors resulting from improper or imprecise condition expressions, | |
| 0 | 0 | Control flow errors - errors resulting from incorrect branching targets, | |
| 0 | 0 | Interface errors - errors resulting from improper passage of data between software modules, | |
| 0 | 0 | Input/Output errors - errors resulting from incorrect data formats or invalid interface specification, | |
| 0 | 0 | Hardware Errors, and | |
| 0 | 0 | Hardware/Software interaction Errors. | |
| 0 | 0 | Task Interaction Errors | |
| 0 | 0 | Other Errors | |
| 0 | 0 | Totals | |

Notes

- a. Performed by the Lead System Engineer for the specific System.
- b. Performed by the HFE team SRO.