MFN 08-269

Enclosure 2

NEDO-33226
Revision 3

ESBWR Licensing Topical Report

ESBWR Software Management Program Manual, Revision 3

Non-Proprietary Version

DO NOT ELECTRONICALLY TRANSMIT TO NRC

HITACHI

GE Hitachi Nuclear Energy

NEDO-33226
Revision 3
Class I
June 2008

Licensing Topical Report

ESBWR - SOFTWARE MANAGEMENT PROGRAM MANUAL

NEDO-33226
Revision 3
DRF#0000-0051-3897
Class I
June 2008

Licensing Topical Report

ESBWR - SOFTWARE MANAGEMENT PROGRAM MANUAL

NON-PROPRIETARY INFORMATION NOTICE

This document is the non-proprietary version of NEDE-33226P, Rev. 3, 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 [[ ]].

IMPORTANT NOTICE REGARDING THE CONTENTS OF THIS REPORT

Please Read Carefully

The information contained in this document is furnished for the purpose of supporting the NRC review of the certification of the ESBWR. The only undertakings of GEH with respect to information in this document are contained in contracts between GEH 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 those participating entities and for any purposes other than those for which it is intended is not authorized; and with respect to any unauthorized use, GEH makes no representation or warranty, and assumes no liability as to the completeness, accuracy, or usefulness of the information contained in this document.

Changes From Previous Revision

Section	Details of Change
Entire document	Renamed the Software Management Plan (SMP) to the Software Management Program Manual (SMPM). This update is reflected throughout the contents of this Manual.
	Reference updated to reflect the renaming of the Software Quality Assurance Plan (SQAP) to the Software Quality Assurance Program Manual (SQAPM). This update is reflected throughout the contents of this Manual.
Section 1	Subsection 1.2 - Added purpose statement.
	Subsection 1.2 - Added Software Management Plan (SMP) to the list of software plans described in the SMPM.
	Subsection 1.2 - Added Software Quality Assurance Plan (SQAP) and Software Test Plan (STP) to the list of software plans described in the SQAPM.
Section 2	Subsection 2.2.4 - Reference to IEEE 603-1991 and correction sheet dated January 30, 1995 was corrected in Section 2.2.4 per the RAI response (7.1-78).
	Subsection 2.3 - The following documents were added to the Supplemental Documents Section: Engineering Computer Programs, Hazardous Business Risk and Safety in GHNEA Services and Products, Hazardous Business Risk Evaluations and Control, and the ESBWR Cyber Security Program Manual.
Section 3	Subsection 3.2 - Added Software Quality Assurance Manager to the list of ESBWR organization functions.
	Subsection 3.2.1 - Added Cyber Security Team information.
	Modified Figure 3-1 to include Cyber Security under I&C/ESE (ACT 43097).
	Configuration Management Manager was relocated from Subsection 3.2.4 (Rev. 2) to Subsection 3.2.3 (Rev. 3).
	Software Quality Assurance Manager was relocated from Subsection 3.2.5 (Rev. 2) to Subsection 3.2.4 (Rev. 3).
	Project Management Team was relocated from Subsection 3.2.6 (Rev. 2) to Subsection 3.2.5 (Rev. 3).
	Training was relocated from Subsection 3.2.3 (Rev.2) to Subsection 3.2.6 (Rev. 3).
	Section 4 Management Process (Rev. 2) was relocated to Section 3 (Rev. 3) and is included as part of the Software Management Plan.
	Subsection 3.6.3 - Additional activities were added to the Project Execution list.
	Subsection 3.9 - Added references to the GEH Project Risk Management

Section	Details of Change
	Procedure, Hazardous Business Risk and Safety in GHNEA Services and Products document, and the Hazardous Business Risk Evaluations and Control document.
	Subsections 4.3.1 Physical Security and 4.3.2 Software Security (Rev. 2) were combined into one Subsection 3.10 Security. References were added to discuss implementation of the security program is defined in the ESBWR Cyber Security Program Plan (ACT 43097).
	Training and Qualifications (Subsection 4.5 in Rev. 2) was relocated to Subsection 3.11 in Rev. 3. Per the response to ACT 43097, applicable cyber security training requirements were added.
Section 4	Section 4 Management Process (Rev. 2) was relocated to Section 3 (Rev. 3) and is included as part of the Software Management Plan.
Section 5	Subsection 5.3 - Revised to address the RAI response (7.1-77) to clarify the use of a modified waterfall model.
	Subsection 5.4, Methods and Tools (Rev. 2) has been divided into two separate subsections, 5.4 Methods and 5.5 Tools (Rev. 3).
	Subsection 5.4.1 Methods (Rev. 2) was combined with Subsection 5.4 Methods (Rev. 3).
	A new Subsection 5.4.7 Cyber Security Analysis has been added to Rev. 3.
	Figures 5-2 through 5-10 have been revised to include Cyber Security activities.
	Figure 5-12 Cyber Security Interaction Model with SMPM and SQAPM Activities has been added to Rev. 3.
	Sections 5.6 through 5.13 including Tables 5.6-1, 5.7-1, 5.8-1, 5.9-1, 5.10-1, 5.11-1, -2, -3, -4, and 5.12-1 have been revised to align with the Figure changes.
	Tables 5.11-2, -3, and –4 have been renamed from Baseline Documents to Output Documents to be consistent with the other Output Document tables.
Section 6	Subsection 6.1.1 Overview (Rev. 2) has been renamed to Subsection 6.1 Introduction (Rev. 3).
	Subsections 6.1.2 through 6.2.2 (Rev. 2) have been renumbered to Subsections 6.2 through 6.10.2 (Rev. 3).
	A new Subsection 6.10.3 Cyber Security Test Engineer has been added to Rev. 3.
	Subsections 6.2.3 through 6.3.4 (Rev. 2) have been renumbered to Subsections 6.10.4 through 6.11.4 (Rev. 3).
	Subsection 6.4 through 6.6 (Rev. 2) have been renumbered to Subsections 6.12 through 6.14 (Rev. 3)

Section	Details of Change
Section 7	Subsections 7.6 through 7.9 of Rev. 2 have been incorporated into Subsection 7.5 of Rev. 3
	Subsections 7.10 through 7.14 of Rev. 2 have been relocated under Subsection 7.6 Methods and Tools of Rev. 3.
	Subsection 7.15 Measurements and Metrics (Rev. 2) has been relocated to Subsection 7.7 Measurements and Metrics (Rev. 3).
Appendix A	Removed internal GEH document numbers and IEEE 603-1998 reference. Updated section to address changes in Sections 5 and 6.
Appendix B	Updated section to be consistent with acronyms used in the SMPM and SQAPM.
Appendix C	Added definition for Criticality Analysis. Revised definitions for Code and Interface.
Appendix F	Revised form to align Section and Error columns.

Contents

List of Tables

List of Figures

1.0 INTRODUCTION

1.1 OVERVIEW

The Software Management Program Manual (SMPM) governs the design and development activities for the Digital Computer-Based I&C software for the ESBWR. Key planning documents for the Instrumentation and Controls (I&C) design team are contained in this manual.

1.2 PURPOSE AND SCOPE

The purpose of the SMPM is to establish the processes and the technical direction for the planning, design, development and management activities of the Digital Computer-Based I&C Software within the scope of the ESBWR Man-Machine Interface (MMI) System and Human Factor & Engineering (HFE) Implementation Plan [2.1].

The scope of the SMPM includes software products with Software Class Q, N3, and N2 (See Appendix C for definitions). Nonsafety-related systems are referenced as Software Class N.

The software plans, identified in the MMIS/HFE IP, and included in the SMPM are:

1. Software Management Plan (SMP) [Section 3.0]
2. Software Development Plan (SDP) [Section 5.0]
3. Software Integration Plan (SIntP) [Section 6.0]
4. Software Installation Plan (SIP) [Section 7.0]
5. Software Operation and Maintenance Plan (SOMP) [Section 8.0]
6. Software Training Plan (STrngP) [Section 9.0]

The ESBWR Software Quality Assurance Program Manual (SQAPM) [2.3(1)] includes the software plans used by Quality Assurance (QA) and the Software Project Engineering (SPE) organizations. The SQAPM governs the same I&C software scope identified in the MMIS/HFE IP. The software plans included in the SQAPM are:

1. Software Quality Assurance Plan (SQAP)
2. Software Safety Plan (SSP)
3. Software Verification & Validation Plan (SVVP)
4. Software Configuration Management Plan (SCMP)
5. Software Test Plan (STP)

Together, the SMPM and the SQAPM 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].

The SMPM shall be in force during all phases of the software life cycle.

The applicable Software Products (software and firmware) covered in the SMPM encompass all I&C systems, as specifically defined in the MMIS/HFE IP [2.1] (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 the SMPM are supplied in Appendix C.

2.0 APPLICABLE DOCUMENTS

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

2.1 SUPPORTING DOCUMENTS

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

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

2.2 REGULATORY DOCUMENTS, CODES AND STANDARDS

The following documents are applicable to the activities specified within the SMPM. The SMPM conforms to planning requirements of these documents except as explicitly noted in Appendix A.

2.2.1 NUREG

- NUREG-0800, Standard Review Plan (SRP), Chapter 7, Branch Technical Position (BTP) HICB-14 R4, Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems

2.2.2 Code of Federal Regulations (CFR)

- 10 CFR 50, 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)

- RG 1.152-2006 - Criteria for Use of Computers in Safety Systems of Nuclear Power Plants
- RG 1.168-2004 - Verification, Validation, Reviews, and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants
- RG 1.169-1997 - Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants
- RG-1.170-1997- Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants
- RG-1.171-1997 - Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants
- RG 1.172-1997 - Software Requirements Specifications for Digital Computer Software Used in Safety Systems of Nuclear Power Plants
- 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 standards are applicable to the activities specified within the SMPM. The SMPM conforms to planning requirements of these standards except as explicitly noted in Appendix A.

The IEEE Standards provide recommended implementation techniques and methods. The SMPM makes specific commitments only to those requirements restated in this document. The ESBWR Project Work Plans shall capture the detailed implementation attributes in accordance with Work Planning and Scheduling [2.3(2.a)]. Future exceptions or deviations from the recommendations specified in the IEEE standards shall require management approval as defined in the SQAPM [2.3(1)] and the SMPM, and are potentially subject to NRC notification in accordance with the MMIS/HFE Implementation Plan [2.1].

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

- IEEE-1008-1987 IEEE Standard for Software Unit Testing
- IEEE-830-1993 IEEE Recommended Practice for Software Requirements Specifications
- IEEE-1074-1995 IEEE Standard for Developing Software Life cycle Processes
- IEEE 603-1991 and correction sheet dated January 30, 1995 - IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations. This IEEE standard is applicable to the design of safety-related instrumentation and control systems of which software is a part.

2.3 SUPPLEMENTAL DOCUMENTS

The following supplemental documents are used in conjunction with the SMPM and enable the performance of the activities stated in Appendix A. These documents are subject to revision to remain current with GEH internal procedures, and do not require the SMPM to be updated when they are revised. Requirements which are being met by these documents shall be maintained via the Requirements Traceability Matrix (RTM).

Reference Number	Document
1.	GE Hitachi Nuclear Energy, "ESBWR - Software Quality Assurance Program Manual," NEDE-33245P, Class III (Proprietary), and NEDO-33245, Class I (Non-Proprietary)

GE Hitachi Nuclear Energy Procedures and Policies		
Reference Number	**Document Title**	**Abstract**
2.a	Work Planning and Scheduling	Defines the process and responsibilities for developing and documenting work plans and schedules for customer-contracted design work and authorized projects. Four key purposes of a Project Work Plan are to define project scope, develop a schedule, monitor progress, and control resources.
2.b	Engineering Computer Programs	Defines requirements for the control of computer programs defined as Engineering Computer Programs.

GE Hitachi Nuclear Energy Procedures and Policies		
Reference Number	**Document Title**	**Abstract**
2.c	Design Review	Defines responsibilities and procedural requirements for conducting formal design adequacy evaluations. Design Reviews are used to verify that product designs meet customer, functional, contractual, safety, health, environmental, regulatory, industry codes and standards, and corporate requirements.
2.d	Design Record File	Defines the process for the generation of a Design Record File, which is a formal, controlled information record for in-progress and completed engineering work.
2.e	Material Request	Details responsibilities and procedural requirements for the release of technical, engineering, customer, and quality requirements that define material, equipment, labor, services and related data to meet GE Hitachi Nuclear Energy (GEH) contract/purchase order, code, and regulatory requirements.
2.f	Independent Design Verification	Details roles and responsibilities for reviewing and substantiating a design to provide independent and documented confirmation that the design meets specified requirements.
2.g	Deferred Design Verification	Defines the process for deferring design verification and for clearing previous deferrals. The process applies to cases where a design, or portion of a design, must be released prior to completion of verification.
2.h	Document Initiation or Change by Engineering Review Memorandum/Engineering Change Notice	Establishes the requirements for the initiation of, or change to, engineering controlled documents by use of the Engineering Review Memorandum/Engineering Change Notice. The process assures traceability, configuration, and quality assurance of engineering documents that are maintained through the current document revision, status, and final disposition.

GE Hitachi Nuclear Energy Procedures and Policies		
Reference Number	**Document Title**	**Abstract**
2.i	Procurement Initiation and Control	Specifies the requirements for procurement of material, equipment, and services, including the application of technical, engineering, customer, and quality requirements on purchase orders. Defines the requirements for establishing and maintaining the Approved Suppliers List.
2.j	Field Deviation Disposition Request	Establishes a process to document and disposition the technical position for field deviations to GE Hitachi Nuclear Energy (GEH) supplied hardware, software, or services. Responsible individuals evaluate Field Deviation Disposition Requests to assure that the proposed field action meets safety, technical, quality, application and commercial requirements.
2.k	Safety-Related Classification	Defines the requirements used to identify structures, systems, components, parts, and technical services that are safety-related. Safety-related structures, systems, components, and parts provide safety-related functions necessary to assure: a. The integrity of the reactor coolant pressure boundary; or b. The capability to shut down the reactor and maintain it in a safe shutdown condition; or c. The capability to prevent or mitigate the consequences of accidents that could result in potential off site exposures comparable to 10CFR50.34(a)(1) or 10CFR100.11 guideline exposures, as applicable.
2.l	Operation and Maintenance Instruction Manuals	Defines requirements applicable to the preparation, review, and approval of Operation and Maintenance instruction manuals.

GE Hitachi Nuclear Energy Procedures and Policies		
Reference Number	**Document Title**	**Abstract**
2.m	Self Assessment, Corrective Action and Audits	Specifies the responsibilities for actions to promptly identify, record and correct Conditions Adverse to Quality to assure that these conditions do not affect the quality of products or services. Defines the requirements and responsibilities for conducting ongoing self-assessments, focused self-assessments, and internal audits of organizations within GE Hitachi Nuclear Energy (GEH).
2.n	Quality and Technical Training	Defines the roles and responsibilities to assure personnel proficiency in quality and technical related activities. The Quality and Technical Training program: a. Assures personnel are trained and proficient in assigned quality and technical tasks. b. Documents qualifications for technical positions, including minimum education, experience, and any special training requirements. c. Records training assignments in a centralized controlled training database.
3.a	Work Authorization	Establishes the requirements and responsibilities within GE Hitachi Nuclear Energy (GEH) for the preparation and approval of Work Authorizations that communicate requirements to functional components of GEH or Global Nuclear Fuel (GNF).
3.b	Project Risk Management Procedure	Implements the project risk management requirements of GE Hitachi Nuclear Energy (GEH) Policy. Provides a controlled process for risk management to maintain positive control of work situations, especially during critical tasks or activities.
3.c	Project Management Policy	Provides requirements for the single Project Management process across all GE Hitachi Nuclear Energy (GEH). The process components include project initiation, planning, scheduling, execution, controls, and post-delivery closeout.

GE Hitachi Nuclear Energy Procedures and Policies		
Reference Number	**Document Title**	**Abstract**
3.d	Project Financial Management	Establishes specific requirements and describes typical methods that are used to assure project financial management activities are accomplished in compliance with GE Hitachi Nuclear Energy (GEH) Policies.
3.e	Quality Policy and Quality System Requirements	Establishes the requirements of the GE Hitachi Nuclear Energy (GEH) business quality system. Defines requirements necessary to implement the quality policy and to demonstrate, by performance both inside and outside GEH, total dedication to the attainment of quality leadership and customer satisfaction.
3.f	Nuclear Energy Quality Assurance Audit Requirements	Establishes the requirements and processes for a comprehensive audit program to verify the implementation and effectiveness of the GE Hitachi Nuclear Energy (GEH) Quality System. The audit program requirements apply to hardware, software and service products and to all personnel who perform quality-related activities on them.
3.g	Reporting of Defects and Noncompliance Under 10CFR Part 21	Defines the requirements and responsibilities within GE Hitachi Nuclear Energy (GEH) for ensuring compliance with the requirements of Part 21 of Title 10 of the Code of Federal Regulations, i.e., 10CFR21, "Reporting of Defects and Noncompliance".
3.h	Hazardous Business Risk and Safety in GHNEA Services and Products	Establishes the organizational responsibilities and systems within GEH to ensure that services and products are evaluated and controlled for hazardous business risks, safety, and environmental effects
3.i	Hazardous Business Risk Evaluations and Control	Defines the responsibilities and practices for evaluation of new GEH activities (products, services, projects, or processes) or changes to existing activities, review of proposed commercial applications, and implementation of risk mitigation and controls.

Reference Number	Document
4.	GE Hitachi Nuclear Energy, "ESBWR Cyber Security Program Plan," NEDE-33295P, Class III (Proprietary), and NEDO-33295, Class I (Non-Proprietary)
5.	GE Hitachi Nuclear Energy, "ESBWR HFE Training Development Implementation Plan," NEDO-33275, Class I (Non-Proprietary)
6.	Institute of Electrical and Electronic Engineers (IEEE), "Standard Glossary of Software Engineering Terminology," IEEE 610.12-1990
7.	Electric Power Research Institute (EPRI), "Guidelines on Evaluation and Acceptance of Commercial Grade Digital Equipment in Nuclear Safety Applications," EPRI TR-106439
8.	GE Hitachi Nuclear Energy, "ESBWR Cyber Security Program Manual," NEDE-33399P, Class III (Proprietary), and NEDO-33399P, Class I (Non-Proprietary)

2.4 ADDITIONAL IEEE STANDARD GUIDANCE

The following IEEE Standards provide additional guidance for the implementation activities. Conformance of the SMPM 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 the SMPM. Clarifications and justifications for such exclusions are provided in Appendix A.

- IEEE-730-2002 - IEEE Standard for Software Quality Assurance Plans
- IEEE-1016-1998 - IEEE Recommended Practice for Software Design Descriptions
- IEEE-1058.1-1987 - IEEE Standard for Software Project Management Plans
- IEEE 1219-1998 - IEEE Standard for Software Maintenance
- IEEE 1228-1994 - IEEE Standard for Software Safety Plans
- IEEE 12207-1996 - IEEE/EIA Standard for Software Life Cycle Processes

2.5 INTERNATIONAL STANDARDS

- ISO 9001:2000, Quality Management Systems-Requirements

3.0 SOFTWARE MANAGEMENT PLAN

3.1 PURPOSE AND SCOPE

The purpose of the Software Management Plan (SMP) is to establish the managerial process and 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].

3.2 ORGANIZATION

The organization is established to address the control of software management and to ensure 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 (I&C/ESE)
2. Software Project Engineering (SPE)
3. Configuration Management Manager (CMM)
4. Software Quality Assurance Manager (SQA Manager)
5. Project Management Team (PMT) (i.e. Project Control)
6. Training

3.2.1 I&C and Electrical Systems Engineering

The I&C software development organization comprises the GEH I&C and Electrical Systems Engineering, the Cyber Security organization and the (GEH and non-GEH) software products vendor organization. The GEH I&C and Electrical System Engineering (I&C/ESE) Organization comprises the I&C and Electrical Systems Engineering Manager (I&C Manager), the platform Technical Project Engineers (TPEs), the Responsible I&C/ESE Engineers and the Cyber Security Team (CyST). This organization implements the activities defined in the SMPM.

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. They are responsible for interfacing with the system engineers and software product vendors. The platform 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 Engineer is responsible for reviewing and confirming that the design

documentation and outputs produced by the software products vendors meet the technical requirements specified in the contract/purchase order.

The software product vendors shall produce the software described in the SMPM. The vendors may be internal or external to GEH and 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 when the Primary POC is unavailable. The Primary 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 the SMPM and the SQAPM [2.3(1)].

The Cyber Security Team (CyST) is responsible for ensuring cyber security of the design, development and evaluation of the Software products throughout the product lifecycle. The CyST is responsible to provide methods to satisfy the cyber security design requirements, methods to assess and validate the actual digital configuration, aid in determining cyber security risks based on the validated configuration and ensure that all necessary cyber security issues are addressed programmatically within the GEH Policy and Procedures to achieve a reasonable level of risk at each ESBWR site for both safety-related and nonsafety-related systems.

[[ ]]

Figure 3-1. Organizational Functions and Interfaces

3.2.2 Software Project Engineering

Software Project Engineer (SPE) is independent of the I&C/ESE organization to ensure organizational freedom to perform quality tasks without undue pressure or conflict of interest related to budget or schedule.

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

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

The SPE organization is described in the SQAPM [2.3(1)], Subsection 3.2.3.4. The Simulation Assisted Engineering (SAE) and Human Factors Engineering (HFE) teams do not perform the quality tasks and are not described in the SQAPM [2.3(1)].

3.2.3 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 6.0].

3.2.4 Software Quality Assurance Manager

The Software Quality Assurance (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 SQAPM [2.3(1)] Subsection 3.2.3.1.

3.2.5 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:

- Project work planning
- Development and maintenance of the integrated project schedule - TPEs shall provide task inputs and support for this activity
- Update of the integrated schedule to show that project tasks are completely and accurately reflected

- Assignment of project resources and skill sets to support the project needs
- Preparation of project progress reports
- Project risk management assessment
- Project budgeting
- Engineering procurement and/or fabrication
- Communication with Licensee and vendors

3.2.6 Training

See Section 3.11 of the SMPM for training requirements.

3.3 ORGANIZATIONAL BOUNDARIES AND INTERFACES

The SMPM and SQAPM 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 subsections:

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/ESE 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 Software Project Engineering Manager

The Software Project Engineering (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 Software Quality Assurance Manager

The Software Quality Assurance (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

The Training Services Lead (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

The Configuration Management Manager (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

The Technical Project Engineer (TPE) has technical responsibility for the software tasks related to software or a group of software products. The TPE's report to the I&C/ESE Manager.

3.5 SOFTWARE MANAGEMENT PLAN CHANGE CONTROL PROCESS

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

The SMPM is a controlled document under configuration control in accordance with the SCMP [2.3(1) Section 6.0].

[[

]]

If a change to the SMPM 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].

Changes to the SMPM require approvals of the following managers or designated appointees: I&C Manager, SPE Manager, and the SQA Manager.

[[

]]

If changes to the SMPM are made, the I&C Manager must document an evaluation indicating that previously completed projects do not have to be reopened to implement the SMPM changes. When changes are made to the SMPM, requirements traceability will be maintained and verified.

3.6 PROJECT MANAGEMENT PRIORITIES, MONITORING, AND CONTROL

The objective of project management is to coordinate the development of project deliverables and to ensure that the deliverables meet the Licensee expectations for nuclear safety, quality, cost, and schedule. The key elements for a successful project delivery by 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 SMPM, the SQAPM [2.3 (1)], 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:

- Project Initiation
- Project Planning and Scheduling
- Project Execution
- Project Controls

- Post-Delivery Closeout

3.6.1 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.

3.6.2 Project Planning and Scheduling

[[

]]

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

[[

]]

The PWP is updated to incorporate changes in the work scope as directed by the PM.

3.6.3 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:

- Developing the Work Breakdown Structure (WBS) for the project.
- Initiating material requisitions for services and materials
- Conducting project kickoff meetings with all interfacing organizations (e.g., Licensee and vendors). The frequency of the project meeting is determined by the PM and shall be conducted with the internal organization (e.g., I&C/ESE and SPE) and the external organization (e.g., Licensee or vendor).
- Conducting a phase Baseline Review (BR) for each software product or logical group of projects at the end of each life cycle phase as described in the SQAPM [2.3(1)].
- Preparing phase Baseline Review Reports in accordance with the SQAPM [2.3(1)].
- Monitoring software product development progress
- Identifying critical path items/activities
- Identifying tools

- Identifying Software Safety Analysis (SSA), Independent Verification and Validation (IV&V), and configuration control activities
- Establishing/reviewing milestone dates for these items/activities
- Identifying and adjusting manpower and resource levels
- Identifying internal and vendor performance problems as early as possible
- Performing Risk Assessment and Risk Management (See Section 3.9)

3.6.4 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 computer-based tools and project reviews. The project review is used to assess the execution of the project.

3.6.4.1 Frequency of Project Review

The frequency of project reviews is commensurate with the complexity of the project. The frequency of project reviews is specified in the PWP.

3.6.4.2 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.

3.6.5 Post-Delivery Closeout

The objective of Post-Delivery Closeout is to finalize the product and to complete the delivery in accordance with the contract. Activities include:

- The closure of project paperwork, including Design Record Files (DRFs)
- Closeout of vendor activities, including vendors submittal of required documentation to GEH
- Turnover of the project to the Licensee, including transfer of SQA activities to the Licensee

The following shall also be performed during Post-Delivery Closeout.

3.6.5.1 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.

3.6.5.2 Software Developed by Vendors

A software product developed by vendors shall conform to the requirements outlined in the SMPM and SQAPM [2.3(1)].

[[

]]

Additionally, Class Q software is approved by the I&C Manager and the SPE Manager and audited by the SQA team.

3.7 METHODS AND TOOLS FOR PROJECT MANAGEMENT

3.7.1 Methods

See Sections 3.6 Project Management Priorities, Monitoring, and Control and Section 3.9 Risk Management for Methods for Project Management.

3.7.2 Tools

The Project Manager shall indicate in the PWP the approved tools required for efficient performance of the project. To ensure efficient and effective execution of the software, the GEH Project Team shall be provided with the tools for project management such as:

- Computers and/or notebooks/laptops
- [[

  ]]
- High speed printers, copiers, and scanners
- Software programs such as Microsoft® Word, Excel, Outlook and Adobe® Acrobat which are widely and commonly used to ensure efficient communication with Licensee and/or vendors

Workforce planning and scheduling tool which allows:

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

Product Data Management System is:

- 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
- The official configuration control system for engineering controlled documents

[[

]]

3.8 BUDGET

[[

]]

The PMT is independent of the engineering teams responsible for the design and quality assurance work on the project. The specific budgetary activities are:

The specific budgetary activities are:

- Accurately allocate resources to each project organization including vendors internal and external to the project organization.
- 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 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 are different from IV&V and SSA activities.

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

A quarterly financial review shall be conducted with the New Plant Project (NPP) General Manager to ensure 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 shall be adjusted to match the cost at completion. In order to achieve the correct cost budget for the project, it is imperative that all costs and commitments are considered when analyzing project costs. 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.

3.9 RISK MANAGEMENT

Risk Management is the process of identifying, controlling, and eliminating or minimizing unpredictable events that may affect the project.

Risk Management shall be implemented in accordance with GEH, Project Risk Management Procedure [2.3(3.b)], Hazardous Business Risk and Safety in GHNEA Services and Products [2.3(3.h)], and Hazardous Business Risk Evaluations and Control [2.3(3.i)].

The Task Leads shall prepare a risk management plan to document responsibilities and actions needed to assess, abate, monitor, and control the identified risks and concerns. It is acceptable that the risk management plan be included in the task-specific PWP.

3.10 SECURITY

Planning and testing to ensure compliance with regulatory cyber security requirements is an integral part of the software development life cycle. Ensuring an adequate Cyber Security Program requires constant changes to ensure protection from new or emerging threats.

To accommodate the need for frequent enhancements to the Cyber Security Program, cyber security requirements will be integrated into the Software Life Cycle through implementing procedures. The design and development of software products shall be performed in accordance with Regulatory Guide 1.152 and ESBWR Cyber Security Program Plan [2.3(4)].

The Cyber Security Program is defined in ESBWR Cyber Security Program Plan [2.3(4)].

3.11 TRAINING AND QUALIFICATION

[[

]]

The Engineering and Project training is performed either by classroom or individual study. The SMPM, SQAPM [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.

[[

]]

4.0 MANAGEMENT PROCESS

Section 4.0 from Rev. 2 has been deleted and the Management Process is now discussed in Section 3.0 as part of the overall Software Management Plan.

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 product's 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] and IEEE 1074 [2.2.4], except as specified in Appendix A.

The purpose of the SDP is to:

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

5.3 ORGANIZATION OF SOFTWARE LIFE CYCLE PROCESS

The software development process follows phase changes in a software life-cycle model.

[[

]]

The software development life cycle is not based on a pure waterfall model, instead it uses a modified waterfall model that includes the provisions for task iteration. The software life cycle phases defined in the SMPM conform to and are based on RG 1.152 [2.2.3], RG 1.173 [2.2.3], and IEEE 1074 [2.2.4]. The software life cycle phases are described as follows:

Planning Phase - In this phase, the definition of the project scope, methodologies, and resources to develop and maintain the deliverable software are 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 Record (BRR) documents successful completion of this phase.

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

Design Phase - In this phase, requirements are transformed into architecture and a detailed representation of software. The Design Phase BRR documents successful completion of this phase.

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

Test Phase - In this phase, the software validation testing occurs which tests for potential defects (errors) and verifies security requirements. The results are documented in the Software Validation Test Report. The Test Phase BRR documents successful completion of this phase.

Installation Phase - In this phase, all activities associated with the installation of the validated software product into the target environment up through final product installation at the plant are conducted, including testing of system security features.

[[

]]

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, cyber security, 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 shall be addressed.

The activities 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

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

5.4.1 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 6.0].

5.4.2 Independent Verification

Independent Verification and Validation (IV&V) shall be conducted on Software Class N and Software Class Q software products.

[[

]]

5.4.3 Testing

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

[[

]]

(See Table 5.7-1 Requirements Phase Outputs Documents or Table 5.8-1 Design Phase Output Documents)

5.4.4 Software Safety Analysis

A Software Safety Analysis (SSA) shall be performed to ensure the safety of Class Q and N3 software. Safety is the most important consideration for the safety-related I&C and takes precedence over budget and schedule. A SSA for software shall be performed according to the SSP [2.3(1) Section 4.0].

[[

]]

5.4.5 Baseline Review

A baseline review is performed for each software product at the completion of each software life cycle phase [2.3(1)].

[[

]]

5.4.6 Deferred Design Verification

Conditional release of a design document may be permissible in 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 5.0].

[[

]]

5.4.7 Cyber Security Analysis

A Cyber Security Analysis (CySA), as defined in CySPP [2.3(4)], shall be performed to ensure the security of Critical Digital Assets (CDAs).

[[

]]

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 both Software Class Q and Software Class N design outputs.

[[

]]

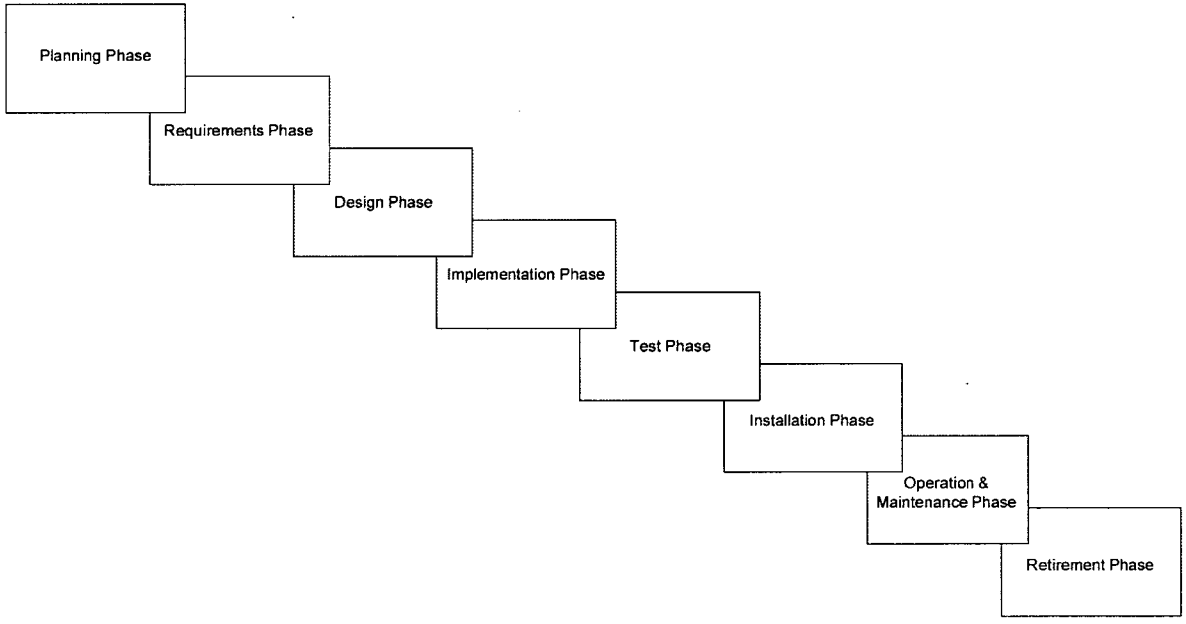

Figure 5-1. Software Life cycle Process Overview

[[ ]]

Figure 5-2. Software Life Cycle Process-Planning Phase

[[ ]]

Figure 5-3. Requirements Phase

[[ ]]

Figure 5-4. Design Phase

[[ ]]

Figure 5-5. Implementation Phase

[[ ]]

Figure 5-6. Test Phase

[[ ]]

Figure 5-7. Installation Phase

[[ ]]

Figure 5-8. Installation Phase (cont-)

[[ ]]

Figure 5-9. Operations and Maintenance Phase and Retirement Phase

[[ ]]

Figure 5-10. Life Cycle Process Notes

[[ ]]

Figure 5-11. Hardware/Software Design Activities

[[ ]]

Figure 5-12. Cyber Security Interaction Model with SMPM and SQAPM Activities

5.6 PLANNING PHASE

[[

]]

5.6.1 Planning Phase Inputs

[[

]]

5.6.2 Planning Phase Outputs

[[

]]

Table 5.6-1 Planning Phase Output Documents

[[			

]]

5.6.3 Software Safety Analysis Report

[[ ]]

5.6.4 Cyber Security Analysis Report

[[ ]]

5.6.5 Planning Phase Baseline Review Record

[[

]]

5.7 REQUIREMENTS PHASE

[[

]]

5.7.1 Requirements Phase Inputs

[[

]]

5.7.2 Requirement Phase Outputs

[[

]]

Table 5.7-1 Requirements Phase Output Documents

[[			
			]]

5.7.3 Requirements Phase Activities

[[

]]

5.7.4 Hardware/Software Specification

[[

(

]]

5.7.5 Software Requirements Specification

[[

]]

5.7.6 System Requirements Specification

[[

]]

5.7.7 Data Communications Protocol

[[

]]

5.7.8 User Interface Specification

[[

]]

5.7.9 Software Support Tools/Documentation for Software Development

[[

]]

5.7.10 Software Safety Analysis Report

[[

]]

5.7.11 Cyber Security Analysis Report

[[ ]]

5.7.12 Requirements Phase Baseline Review Record

[[

]]

5.8 DESIGN PHASE

[[

]]

5.8.1 Design Phase Inputs

[[

]]

5.8.2 Design Phase Outputs

[[

]]

Table 5.8-1 Design Phase Output Documents

[[			
			]]

5.8.3 Design Phase Activities

5.8.3.1 Software Design Description

[[

]]

5.8.3.2 Intra System Communication Protocol Specification

[[

]]

5.8.3.3 Software Coding Conventions and Guidelines Document

[[

]]

5.8.3.4 Software Support Tool Documentation Package

The Software developer shall evaluate the use of software tools in the new design and document the intended use consistent with the Software Requirements Specification and Software Design Description.

5.8.3.5 Application of Previously Developed Software

[[

]]

The Software developer shall integrate the PDS package into the new design and document this integration consistent with the Software Requirements Specification and Software Design Description.

5.8.3.6 Commercial Off-The-Shelf Software

[[

]]

5.8.3.7 Software Validation Test Plan, Procedures, and Test Cases Specification

[[

]]

5.8.3.8 Site Acceptance Test Procedure Development

[[

]]

5.8.3.9 Multi-System Factory Acceptance Test Procedure Development

[[

]]

5.8.3.10 System Factory Acceptance Test Plans and Procedures

[[

]]

5.8.3.11 Software Safety Analysis Report

[[

]]

5.8.3.12 Cyber Security Analysis Report

[[ ]]

5.8.3.13 Design Phase Baseline Review Record

[[

]]

5.9 IMPLEMENTATION PHASE

[[

]]

5.9.1 Implementation Phase Inputs

[[

]]

5.9.2 Implementation Phase Outputs

[[

]]

Table 5.9-1 Implementation Phase Output Documents

[[			
			]]

5.9.3 Implementation Phase Activities

5.9.3.1 Software Coding Readiness Review

[[

]]

5.9.3.2 Software Coding

[[

]]

5.9.3.3 Code Review

[[

]]

5.9.3.4 Software Functional Testing

[[

]]

5.9.3.5 Software Functional Test Report

[[

]]

5.9.3.6 Software Build Description

[[

]]

5.9.3.7 Finalize Software Validation Test Plan, Procedures, and Test Cases Specification

[[

]]

5.9.3.8 Software Safety Analysis Report

[[

]]

5.9.3.9 Cyber Security Analysis Report

[[

]]

5.9.3.10 Implementation Phase Baseline Review Record

[[

]]

5.10 TEST PHASE

[[

]]

5.10.1 Test Phase Inputs

[[

]]

5.10.2 Test Phase Outputs

[[

]]

Table 5.10-1 Test Phase Output Documents

[[			
			]]

5.10.3 Software Validation Testing

[[

]]

5.10.4 Software Validation Test Report

[[

]]

5.10.5 Production Release

[[

]]

5.10.6 Software Release Notes

[[

]]

5.10.7 Cyber Security Analysis Report

[[

]]

5.10.8 Human Factors Engineering Verification and Validation

[[

]]

5.10.9 Test Phase Baseline Review Record

[[

]]

5.11 INSTALLATION PHASE

[[

]]

5.11.1 Installation Phase Inputs

[[ ]]

Table 5.11-1 Installation Phase Input Documents

[[		

]]

5.11.2 Installation Phase Outputs

[[

]]

Table 5.11-2 Installation Phase I Output Documents

[[			
			]]

Table 5.11-3 Installation Phase II Output Documents

[[			
			]]

Table 5.11-4 Installation Phase III Output Documents

[[			
			]]

5.11.3 System Factory Acceptance Testing

[[

]]

5.11.4 Multi-System Factory Acceptance Testing

[[

]]

5.11.5 Site Acceptance Testing

[[

]]

5.11.6 Software Operations & Maintenance Manuals

[[ ]]

5.11.7 Software Training Manuals

[[ ]]

5.11.8 HFE, ISV, V&V, Result Summary Report

Initially, Human Factor Engineers work with the static performance information of the simulator equipment.

[[

]]

Human Factor Engineers take this information and adjust the settings in the simulator as needed.

5.11.9 Cyber Security Analysis Report

[[

]]

5.11.10 Installation Phase Baseline Review Record

[[

]]

5.12 OPERATIONS AND MAINTENANCE PHASE

[[

]]

5.12.1 Operations and Maintenance Phase Inputs

[[

]]

5.12.2 Operations and Maintenance Phase Outputs

[[ ]]

Table 5.12-1 O&M and Retirement Phase Output Documents

[[			
			]]

5.12.3 Operations and Maintenance Activities

[[

]]

5.12.4 Cyber Security Analysis Report

[[

]]

5.12.5 Operations and Maintenance Phase Baseline Review Record

[[

]]

5.13 RETIREMENT PHASE

[[

]]

5.13.1 Retirement Phase Activities Baseline Review Record

[[

]]

6.0 SOFTWARE INTEGRATION PLAN

6.1 INTRODUCTION

The Software Integration Plan (SIntP) consists of three major phases: integrating the various software modules together to form single programs, integrating the software module integration result with the hardware and instrumentation, and testing the resulting integrated product. During the first phase, the various object modules are combined to produce executable programs. In the second phase, these programs are then loaded 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 of the product integration produced during the second phase.

Software integration shall be performed during the software functional test.

6.2 PURPOSE

The purpose of the SIntP is to:

- Describe the organization and responsibilities of individuals or groups involved in the SFT activities.
- Describe software functional test management (e.g., 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 SMPM shall be justified and approved by the RTPE. The justification and approval shall be prepared, reviewed, approved, and maintained in the software project DRF.

6.3 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.

During software integration, interface analysis, data flow analyses, timing, and sizing analysis shall be performed as appropriate. The results of the analyses shall be documented in the Software Functional Test Data Sheet (See Appendix D).

6.4 ORGANIZATION AND MANAGEMENT

The Software Functional Test is performed by the Software Development organization. Section 6.10 describes the test personnel roles and responsibilities. SVVP [2.3(1), Section 5.0] describes the IVVT roles and responsibilities. CySPM [2.3 (8)] describes the Cyber Security roles and responsibilities.

6.5 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 and CyST. The CyST is responsible for performing Cyber Security Analysis on the Software Functional Test Report (SFTR). The IVVT is responsible for performing the IV&V on the class Q Software Functional Test Report (SFTR).

6.6 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.

6.7 RESOURCES

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

- Test facilities
- Test equipment and tools
- Qualified test engineers
- Any special needs for security, including cyber security adherence to CySPM [2.3 (8)]

6.8 TRAINING

The Cyber Security Engineering Leader, IVVT Task Lead, and the RTPE shall ensure that their staff is trained to support test activities.

6.9 REVIEWS

The progress of testing and issues related to testing shall be evaluated on a regular basis (e.g., 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 given to circumstances indicating deficiencies in the testing process. If a deficiency is identified, the CAR process, as defined in Section 5.1.4, shall be used to initiate corrective actions to improve the test process.

6.10 TEST PERSONNEL ROLES AND RESPONSIBILITIES

This section defines the test personnel roles and responsibilities.

6.10.1 Responsible Technical Project Engineer

The Responsible Technical Project Engineer (RTPE), as part of the design team, has technical responsibility for the software functional test tasks.

6.10.2 Software Functional Test Engineer

The Software Functional Test Engineer (RTE) is responsible for designing, executing, and documenting the test results in accordance with the SDD intra-system data communication protocol specification and the SMPM.

6.10.3 Cyber Security Test Engineer

The Cyber Security Test Engineer (CySTE) is responsible for designing and executing the adversary based test plans, procedures, and test cases. The CySTE shall document the test results.

6.10.4 Test Personnel Qualifications

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

6.11 SOFTWARE FUNCTIONAL TEST GUIDELINES

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

- Test preparation

- Test design
- Test execution
- Test summary

This test guideline conforms to RG 1.170 [2.2.3] and IEEE 829 [2.2.4].

6.11.1 Test Preparation Guidelines

The purpose of test preparation is to ensure that the required test activities are 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. For Class Q and N3 software, QA shall review procedures and identify hold points, as required, prior to final procedure approval.

The individual responsible for test preparation shall perform the following tasks:

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

6.11.2 Test Design Guidelines

The test designer shall perform the following tasks:

- Specify the software features to be tested for each software item.
- Specify and provide justification for the software features not to be tested (e.g., previously tested unmodified features and modifications have been demonstrated to not require re-test of these features).
- Determine the test approach and specify the test techniques.

- Specify the test cases and acceptance criteria for each item.
- 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 using requirements external to a feature to derive test cases and test procedures. Functional testing verifies the end results at the feature I/O level. However, functional testing 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 conducted to evaluate a system or component 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 ensure that all test features are identified for software class Q.

6.11.3 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:

- Obtain test items including relevant reference documentation.
- Set the test environment.

- Observe the software and hardware during testing for both the expected and unexpected behaviors.
- Confirm completeness and test termination requirements are satisfied.
- Document the test results while executing the test procedures.
- Initiate the change control process per SCMP [2.3(1), Section 6.0] to resolve design errors encountered during the test.

6.11.4 Test Summary Guidelines

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

- Test activities
- Test results
- Requirement traceability
- 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 speciation have been verified by the test. The SQAPM [2.3(1)] provides methods for performing traceability analysis.

6.12 METHODS

Methods used to perform software testing are described in the following sections.

6.12.1 Software Functional Testing (Module Level)

[[

]]

6.12.1.1 Module/Unit Test Preparation

[[

]]

6.12.1.2 Module Test Design

[[

]]

6.12.1.3 Module Test Execution

[[

]]

6.12.1.4 Module Test Summary

[[

]]

6.12.2 Software Functional Testing (Integration Level)

[[

]]

6.12.2.1 Integration Test Preparation

[[

]]

6.12.2.2 Integration Test Design

[[

]]

6.12.2.3 Integration Test Execution

[[

]]

6.12.2.4 Software Functional Test Summary

[[

]]

6.12.3 Software Validation Testing

[[

]]

6.12.3.1 Software Validation Testing - Internal I&C/ESE

6.12.3.1.1 Validation Test Preparation

[[

]]

6.12.3.1.2 Validation Test Design

[[

]]

6.12.3.1.2.1 Instrument Level Validation Test Design

[[

]]

6.12.3.1.2.2 System Level Validation Test Design

[[

]]

6.12.3.1.2.3 Validation Test Execution

[[

]]

6.12.3.2 Software Quality Assurance Software Validation Testing

Software Quality Assurance Software Validation Testing is described in the SQAPM [2.3(1)].

6.13 TEST DOCUMENTATION AND PROBLEM REPORTING

[[

]]

6.13.1 Software Functional Test Documentation

6.13.1.1 Software Functional Test Data Sheet

[[

]]

6.13.1.2 Software Functional Test Metrics Sheet

[[ ]]

6.13.1.3 Software Functional Test Report

[[

]]

6.14 MEASUREMENT AND METRICS

[[ ]]

7.0 SOFTWARE INSTALLATION PLAN

7.1 INTRODUCTION

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

7.2 PURPOSE

The purpose of the SIP is to:

- Define the installation phase activities.
- Describe the installation procedures.
- 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.
- Provide the requirements and guidelines necessary to prepare, execute, and document software installation.

7.3 SCOPE

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

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.

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.

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:

- A description of the software installation procedure
- Software installation methods and procedures
- Criteria used to determine the success or failure of the installation effort
- A checklist or sequence of steps that can be used to confirm that correct software is installed in the specific systems in accordance with the system design documents. The following is a sample list of items to be considered as part of the checklist:
    - Affected functions are inoperable and in a safe condition according to the plant's technical specifications before proceeding with installation.
    - The computer system is functional.
    - The sensors and actuators are functional.
    - All cards are present and installed in the correct slots.
    - The communication system is correctly installed.
    - The correct software versions are installed on the correct computers.
    - Appropriate return-to-service testing has been successfully conducted before declaring the modified function operable.
    - Installation configuration tables are complete.
    - Environmental conditions (e.g., temperature, humidity, vibration, and rack space) are considered and provided for.
    - Special tools, methods, or techniques used to accomplish the installation function shall be identified.
    - Installation tools shall be qualified with a degree of rigor and level of detail appropriate to the safety significance of the software utilizing the installation tools.
    - Security provisions have been satisfied.
    - Precautions to ensure personnel and plant safety have been identified.

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. However, each system or logical group of systems should have its own installation report.

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

An installation report for each package or system shall be produced upon the completion of the installation effort. The installation report shall include the following installation activities as a minimum:

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

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 the software is correctly configured for the operating safety system.

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

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. Each configurable mode shall include the function, safety, and security of the overall application. The configuration tables shall include the following items:

- 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.

- Installation configuration tables shall be consistent with the software specifications as described in the SRS, SDD, software code, and software build documents.
- Software configuration tables shall contain system specific data.
- 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.

7.5.4 Operations and Maintenance Manuals

The software Operations and Maintenance (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.

Operations and Maintenance Manuals are described in Subsection 8.5.1.

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 O&M manuals. The software training manuals provide the basis for training the Licensee or end-user.

Training manuals are described in Subsection 9.3.1.

7.6 METHODS AND TOOLS

7.6.1 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.

7.6.2 Software Archive Retrieval

The software package shall be placed under CM as required by the SCMP [2.3(1) Section 6.0]. Plant-specific methods of archival and retrieval are the responsibility of the Licensee and are beyond the scope of the SMPM. Where applicable, specific backup and recovery procedures shall be included in the maintenance section of the O&M manual.

7.6.3 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.

7.6.4 Installation Documentation and Problem Reporting

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

7.6.5 Verification and Validation Methods

The installation phase outputs shall be verified in accordance with the SVVP, [2.3(1) Section 5.0].

7.7 MEASUREMENTS AND METRICS

Measurement and metrics shall be developed in accordance with the SQAPM [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

The SOMP defines requirements, methods, and considerations for developing the system O&M Manual. The SOMP 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 the SOMP 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 SQAPM [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 Subsection 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. These activities are 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 including:

- Monitoring the software to detect security breaches. This includes penetration or attempted penetration of the system.
- Measuring, recording, evaluating, analyzing, reporting system errors and error rates, and determining root cause. This shall include consideration for cyber security incidents.

- Surveillance procedures for ensuring that:
  - The system is operating correctly and is calibrated.
  - The software state is consistent with the plant-operating mode.
- The system is ready and able to perform its safety-related function, for all Class Q software.
- The system is ready and able to control and monitor plant operation, for Class N software.
- Developing backup and restore procedures for configuration, data, and code.
- Developing calibration procedures.
- Developing maintenance procedures.
- Developing cyber security risk mitigation procedures in accordance with the CySPM [2.3 (8)].

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 Configuration Items (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 GEH and the Licensee shall meet 10 CFR 50 Appendix B and 10 CFR Part 21. Both GEH and the Licensee will have a quality assurance program and a 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 Phase procedures shall include:

- A description of the responsibilities and authority of the operators
- A description of the security requirements for operating the software system taking into consideration Safeguards imposing restricted access in accordance with the CySPM [2.3 (8)]
- 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. Contingency plans shall be created to ensure appropriate response to penetration.

8.4.2 Maintenance Phase Procedures

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

- A description of the method used for software risk management during maintenance. Particular attention shall be given 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 Phase 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.

Each O&M manual shall be developed in accordance with Operation and Maintenance Instruction Manuals [2.3.2 (l)].

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

- The operating modes
- Error messages including description and error recovery methods
- Backup and recovery procedures
- Operator actions shall be specified in terms of inputs supplied by the operator or system
- Actions initiated by the operator
- 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. This includes:

- Precautions
- Limitations
- Personnel or plant hazards
- Maintaining the integrity of the Cyber Security Defensive Model including considerations for restricted access to the model itself in accordance with the CySPM [2.3(8)]
- Variables in the physical environment that the software must monitor and control
- User interfaces. User interfaces shall be described fully for each category of operator or user.
- Required actions to ensure cyber security protection during O&M, restoration of integrity of cyber security defensive model after O&M activities, and recognition of cyber events in which indication, control, and protection features may have been compromised.

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 of the O&M manual shall include:

- Precautions
- Limitations
- Personnel or plant maintenance hazards
- Security vulnerabilities
- Trouble shooting and reporting procedures and methods
- A description of or reference to Configuration Management and Change Control procedures. The Configuration Management and Change Control procedures shall:
    - 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.
    - Ensure that software is functioning properly after the maintenance.
    - Upgrade field procedures. Field upgrade procedures shall be described, including:
        - Installation procedures
        - Installation test procedures
        - Installation test checklists
        - Installation test data sheets

8.5.1 Software Operation and Maintenance Manuals

The Software Operation and Maintenance Manual shall be developed in accordance with the requirements outlined in both 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:

- Information necessary for all operating modes. This includes normal operation, off normal operation, and emergency operation.
- Start-up and shutdown of the software product. This includes error recovery and backup.
- A list of error messages. Error messages shall be listed together and include definitions and corrective action(s) by the operator.
- Description of the operational environment within which the software shall operate. This description shall include precautions and limitations that must be observed during operations to avoid exposing personnel or the plant to hazards or security vulnerabilities.

- Description of each user interface for each category of user, including operators, shift supervisors and, nuclear engineers.

The Software Maintenance Manual shall, at a minimum, include:

- Description of the procedures describing when operational software must be changed
- Identification of the precautions and limitations that must be observed during maintenance to avoid exposing personnel or the plant to hazards or security vulnerabilities.
- Change control process - Configuration Management shall be described or referenced.
- Software Installation Procedure - This includes regression test steps that confirm the revised/enhanced software is correctly installed and that no faults have been introduced in the revised/enhanced software.
- Methods used to restore older versions of software and methods used to back up software.
- Methods to troubleshoot and diagnose both the system and the 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 5.0].

8.6 MEASUREMENT AND METRICS

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

9.0 SOFTWARE TRAINING PLAN

9.1 INTRODUCTION

The 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].

9.1.1 Purpose

The purpose of the STrngP is to define:

- The requirements and methods used in 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 STrngP 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. The training requirements include proper usage (e.g., personal safety, system security) of the equipment for the users, operators, maintenance personnel, and management personnel. The SMPM 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 GEH Nuclear Services. The organizational responsibilities are identified in Subsection 3.4. The TSL is a functional position responsible for assignment of personnel to support training for the software products. The Training Services Lead (TSL) is responsible for ensuring 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 the 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 including:

- Development and maintenance of training plans.
- Development and review of the training manual.
- Development of training courses.
- 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:

- Startup
- Shutdown
- Installation
- Backup
- Restoration
- Configuration
- Calibration
- Troubleshooting
- Replacing failed modules
- Plant modes, including alarm and indicator responses
- Training assessment
- Operating specific scenarios
- Recommended surveillance testing
- Security which includes Cyber Security

9.3.1 Software Training Manual Program

Software training manuals shall include the following requirements:

- Description of actions available to the operator and the technician for all operating modes, including error recovery.
- Description of operator actions specified in terms of inputs supplied by users and equipment, actions initiated by the operation, and responses to the user.
- 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.
- 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.
- 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 specified in the contract. Software training manuals are described in Subsection 9.6.1.

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 the content and methods for each training course. 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 provides the most effective nuclear training. Examples of effective training facilities are:

- Dedicated classroom space (e.g., conference rooms)
- Instructor-led classroom software lab facilities
- Self-study computer lab facilities
- A remote training access tool (e.g., presentation tools) which allow training at a remote training workstation
- Control room simulator

9.7 MEASUREMENT AND METRICS

Metrics provide a basis for determining the effectiveness of the training program. Metrics selected during the development of the training program may be a combination of tools based on the nature of the training program being offered. For example, a training course providing a one-day overview session would utilize a different set of metrics than a four-week course that utilizes extensive use of simulation training tools. The training 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 are:

- Instructor Assessment. The instructor queries trainees during class session and grades the daily performance of the trainees.
- 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.

10.0 APPENDICES

10.1 APPENDIX A SOFTWARE PLANS CONFORMANCE REVIEW

The Regulatory Guides and IEEE Standards have been reviewed for conformance. In general, the IEEE Standards provide more detailed guidance for the implementation activities. When requirements derived from the Standards are specifically addressed within this plan, a commitment to the approach is made. Conformance clarification and justification is provided in this Appendix.

[[	
	]]

Appendix A - Software Plans Conformance Review								
Item	Reg Guide	IEEE Stand.	Related Software Plan			Deviation	Conform. Code	Justification
			SMPM	SQAPM	None			
[[								
								]]

Appendix A - Software Plans Conformance Review								
Item	Reg Guide	IEEE Stand.	Related Software Plan			Deviation	Conform. Code	Justification
			SMPM	SQAPM	None			
Regulatory Guides								
[[								
							]]	

Appendix A - Software Plans Conformance Review								
Item	Reg Guide	IEEE Stand.	Related Software Plan			Deviation	Conform. Code	Justification
			SMPM	SQAPM	None			
IEEE Standards from Section 2.2.4								
[[								

Appendix A - Software Plans Conformance Review								
Item	Reg Guide	IEEE Stand.	Related Software Plan			Deviation	Conform. Code	Justification
			SMPM	SQAPM	None			

Appendix A - Software Plans Conformance Review								
Item	Reg Guide	IEEE Stand.	Related Software Plan			Deviation	Conform. Code	Justification
			SMPM	SQAPM	None			

Appendix A - Software Plans Conformance Review								
Item	Reg Guide	IEEE Stand.	Related Software Plan			Deviation	Conform. Code	Justification
			SMPM	SQAPM	None			

Appendix A - Software Plans Conformance Review								
Item	Reg Guide	IEEE Stand.	Related Software Plan			Deviation	Conform. Code	Justification
			SMPM	SQAPM	None			

Appendix A - Software Plans Conformance Review								
Item	Reg Guide	IEEE Stand.	Related Software Plan			Deviation	Conform. Code	Justification
			SMPM	SQAPM	None			
								]]

Appendix A - Software Plans Conformance Review								
Item	Reg Guide	IEEE Stand.	Related Software Plan			Deviation	Conform. Code	Justification
			SMPM	SQAPM	None			
IEEE Standards from Section 2.4								
[[								

Appendix A - Software Plans Conformance Review								
Item	Reg Guide	IEEE Stand.	Related Software Plan			Deviation	Conform. Code	Justification
			SMPM	SQAPM	None			

Appendix A - Software Plans Conformance Review								
Item	Reg Guide	IEEE Stand.	Related Software Plan			Deviation	Conform. Code	Justification
			SMPM	SQAPM	None			
								]]

10.2 APPENDIX B ACRONYMS AND ABBREVIATIONS

The following acronyms and abbreviations are used throughout the SMPM.

Acronym	Meaning
AOF	Allocation of Function
ASL	Approved Suppliers List
ASME	American Society of Mechanical Engineers
ATWS	Anticipated Transients without Scram
BR	Baseline Review
BRR	Baseline Review Record
BRT	Baseline Review Team
BTP	Branch Technical Position (see HCIB)
CAQ	Condition Adverse to Quality
CAR	Corrective Action Request
CCB	Change Control Board
CEO	Chief Executive Officer
CFR	Code of Federal Regulations
CI	Configuration Item
CM	Configuration Management
CMM	Configuration Management Manager
CMS	Configuration Management System
COTS	Commercial-Off-The-Shelf
CySPP	Cyber Security Program Plan
CTS	Commitment Tracking System
DCD	Design Control Document
DCPS	Data Communication Protocol Specifications

Acronym	Meaning
DRF	Design Record File
ECA	Engineering Change Authorization
ECN	Engineering Change Notice
EIA	Electronic Industries Alliance
EMC	Electromagnetic Compatibility
EOP	Engineering Operating Procedure
EPRI	Electrical Power Research Institute
ERM	Engineering Review Memorandum
ESBWR	Economic Simplified Boiling Water Reactor
FDDR	Field Deviation Disposition Request
FDI	Field Disposition Instruction
FMEA	Failure Modes and Effects Analysis
FRA	Functional Requirements Analysis
GE	General Electric Company
GEH	GE Hitachi Nuclear Energy
HFE	Human Factors Engineering
HFEITS	Human Factors Engineering Issue Tracking System
HICB	Instrumentation and Control Branch, NRC Branch Technical Positions for I&C
HIS	Human System Interface
HSS	Hardware/Software Specification
ICPS	Intra-system Communication Protocol Specification
I&C	Instrumentation and Controls
I&C ESE	Instrumentation and Controls Electrical Systems Engineering
IEEE	Institute of Electrical and Electronic Engineers
I/O	Input/Output

Acronym	Meaning
IP	Implementation Plan
ISO	International Standards Organization
IV&V	Independent Verification and Validation
IVVT	Independent Verification and Validation Team
LD	Logic Diagram
LLC	Limited Liability Corporation
LTR	Licensing Topical Report
MCR	Main Control Room
[[	]]
MMI	Man Machine Interface
MMIS	Man Machine Interface System
N/A	Not Applicable
N-DCIS	Non Safety – Distributed Control and Information System
NPP	New Plant Project
NRC	Nuclear Regulatory Commission
O&M	Operation and Maintenance
P&ID	Piping & Instrumentation Diagram
P&P	Policies and Procedure
PDM	Project Design Manual
PDMS	Product Data Management System
PDS	Previously Developed Software
PM	Project Manager
PMT	Project Management Team
POC	Point of Contact
PQC	Product Quality Certification

Acronym	Meaning
PR	Problem Report
PRA	Probabilistic Risk Assessment
PWP	Project Work Plan
Q-DCIS	Distributed Control and Information System, Safety Related Portion, see N-DCIS
QA	Quality Assurance
QCE	Quality Control Engineer
RCCE	Responsible Configuration Control Engineer
RE	Responsible Engineer
RG	Regulatory Guide
RMCN	Review Memorandum Change Notice
RSE	Responsible System Engineer
RSR	Results Summary Report
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
SBD	Software Build Description
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SDD	Software Design Description

Acronym	Meaning
SDP	Software 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 Installation Test Team
SMP	Software Management Plan
SMPM	Software Management Program Manual
SOMP	Software Operations and Maintenance Plan
SPE	Software Project Engineering
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
SQAPM	Software Quality Assurance Program Manual
SRP	Standard Review Plan
SRS	Software Requirements Specification
SSA	Software Safety Analysis
SSP	Software Safety Plan
SST	Software Safety Team
STP	Software Test Plan
STrngP	Software Training Plan
SVT	Software Validation Testing
SVVP	Software Validation and Verification Plan

Acronym	Meaning
SyRS	System Requirement Specification
TA	Task Analysis
TBD	To Be Determined
TPE	Technical Project Engineer
TR	Topical Report
TSL	Training Services Lead
UIS	User Interface Specification
V&V	Verification and Validation
WBS	Work Breakdown Structure

10.3 APPENDIX C 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

Term	Definition
	provide [IEEE 610.12].
Certification	A written guarantee that a system or component complies with its specified requirements and is acceptable for operational use [IEEE 610.12].
Code	Review of software source codes to identify coding errors and to verify that software design as specified in the Software Design Description been correctly and completely implemented.
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. GEH Self-Assessment, Corrective Action and Audits [2.3(2.v)] 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 of 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].
Criticality Analysis	The degree of impact that a requirement, module, error, fault, failure, or other item has on the development or operation of a system. A method used to determine the impact of the software product on the system & environment as a whole and thereby determine the software importance (i.e. Software Class Q, N3 or N2).
Design	Design Documentation is information recorded about a specific life

Term	Definition
Documentation	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.
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.
Design phase	The *phase* 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 GEH 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 GEH Design Review [2.3(2.e)]. 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

Term	Definition
	deviation required in the field in supplied hardware, software, or services (see GEH Field Deviation Disposition Request [2.3(2.q)]).
Firmware	The combination of a hardware device and computer instructions and data that reside as read-only software on that device [IEEE 610.12].
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 *phase* 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 *phase* 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 an embedded microprocessor(s).
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	1) A shared boundary across which information is passed. This definition is interpreted broadly to include design interfaces between participating design organizations. 2) A hardware or software component that connects two or more other components for the purpose of passing information from one to the other. 3) To connect two or more components for the purpose of passing information from one to the other. 4) To serve as a connecting or connected component as in (2). [IEEE 610.12 as modified by RG 1.69
Metric	A quantitative measure of the degree to which a system, component, or process possesses a given attribute [IEEE 610.12].

Term	Definition
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 *phase* 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 *phase* 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.
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 Management 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].
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	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]. 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). Requirements are not specified with the word **should.** Instead, it 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

Term	Definition
	necessarily required; Also, that (in the negative form) a certain course of action is not prohibited ('should' and 'recommended' are equivalent in meaning).
Requirements Phase	The *phase* in the software life cycle during which the requirements for a software product are defined and documented [IEEE 610.12].
Requirements Traceability Analysis	The process of studying user needs to arrive at a definition of system, hardware, or software requirements [IEEE 610.12].
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 GEH Independent Design Verification [2.3(2.i)] for verifications, or in GEH Deferred Design Verification [2.3(2.j)] 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 Class N2	Nonsafety-related system software whose failure cannot adversely affect a safety related function.
Software Class N3	Nonsafety-related systems software whose failure could challenge safety systems as defined below: a. Software whose inadvertent response to stimuli, failure to respond when required, response out-of-sequence could result in a accident or transient as defined in the DCD, Chapter 15 [2.1(6)]. 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	Software performs functions classified per GEH Safety-Related Classification determination process [2.3(2.t)] as Safety-Related.
Software Development Process	The process by which user needs are translated into a software product. The process involves translating user needs into software

Term	Definition
	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 eight *phases*, namely, Planning, Requirements, Design, Implementation, Integration, Validation, Installation, and Operation & Maintenance.
Software Module	See Module
Software Package	See Package
Software Unit	See Module
Source Code	Computer instructions and data definitions expressed in a form suitable for input to an assembler, compiler, or other translator.
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 plan. These are the enabling documents that either augment or enable the performance of the activities stated in this plan.
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 plan. These documents form the design basis for the activities stated in this plan.
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

Term	Definition
	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 *phase* 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 Module Testing	Testing of individual hardware or software units or groups of related units [IEEE 610.12].
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 GEH Design Review [2.3(2.e)] or GEH Independent Design Verification [2.3(2.i)] 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 Software 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.

10.4 APPENDIX D SOFTWARE FUNCTIONAL TEST DATA SHEET (EXAMPLE)

SOFTWARE FUNCTIONAL TEST DATA SHEET DRF#

Page: of

Responsible Engineer:		**Page:** of
Integration Test Engineer:	**Sign:**	**Date:**

Application S/W Package:	**Rev:**	**Software Class:**
Location:	**Instrument:**	
Test Entity:		

Test Description	**Finding**	**Status**

10.5 APPENDIX E SOFTWARE FUNCTIONAL TEST METRICS SHEET (EXAMPLE)

SOFTWARE FUNCTION TEST METRICS SHEET DRF#

Page: 1 of 1

Application Software Package: Revision:

Subsystem: Software Class:

Error Type	***SFT Section***	***Total Errors***
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

10.6 APPENDIX F 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