

**MFN 08-256**

**Enclosure 2**

**NEDO-33245  
Revision 3**

**ESBWR Licensing Topical Report – ESBWR Software  
Quality Assurance Plan Manual**

**Non-Proprietary Version**

**DO NOT ELECTRONICALLY TRANSMIT TO NRC**



**HITACHI**

*GE Hitachi Nuclear Energy*

---

NEDO-33245  
Revision 3  
Class I  
July 2008

**Licensing Topical Report**

**ESBWR - Software Quality Assurance  
Program Manual**

NEDO-33245, Rev. 3

---

NEDO-33245  
Revision 3  
DRF#0000-0049-7144  
Class I  
July 2008

**Licensing Topical Report**

**ESBWR - Software Quality Assurance  
Program Manual**

**NON-PROPRIETARY INFORMATION NOTICE**

This document is the non-proprietary version of NEDE-33245P, 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.

Copyright 2008, GE-Hitachi Nuclear Energy Americas LLC, All Rights Reserved.

## **Changes From Previous Revision**

### **Revision 3 Summary of Record of Changes**

**Structure Changes:** Revision 3 has been streamlined to 8 sections from 18 sections in Revision 2. While the content remains the same, sections have been appropriately rearranged within the SQAPM to provide more efficient organization and navigation.

- Section 3.0 is the Software Quality Assurance Plan (SQAP) (Revision 2, Section 3.0 was Management Organization and this Plan was not listed)
  - Subsection 3.2 is Management Organization
  - Subsection 3.3 is Documentation (Revision 2, Section 4.0)
  - Subsection 3.4 is Standards, Practices, Conventions and Metrics (Revision 2, Section 5.0)
  - Subsection 3.5 is Reviews and Audits (Revision 2, Section 6.0)
  - Subsection 3.6 is Problem Reporting and Corrective Action (Revision 2, Section 11.0)
  - Subsection 3.7 is Tools, Techniques, and Methodologies (Revision 2, Section 12.0)
  - Subsection 3.8 is Code and Media Control (Revision 2, Section 13.0)
  - Subsection 3.9 is Vendor and Acquired Software Control (Revision 2, Section 14.0)
  - Subsection 3.10 is Records, Collection, Maintenance, and Retention (Revision 2, Section 15.0)
  - Subsection 3.11 is Training (Revision 2, Section 16.0)
  - Subsection 3.12 is Risk Management (Revision 2, Section 17.0)
- Section 4.0 is the Software Safety Plan (SSP) (Revision 2, Section 9.0)
- The following new sections were added to this plan:
  - Qualifications and Training
  - Software Life Cycle
  - Software Safety Program Records

- Software Configuration Management Activities
- Software Quality Assurance Activities
- Software Verification and Validation Activities
- Subcontract Management
- The following sections had additional content added:
  - Organization and Responsibilities
  - Documentation Requirements
  - Tool Support and Approval
  - Previously Developed or Purchased Software
  - Process Certification
- Revised Subsection 4.3.1 through 4.3.7.1 (Revision 2, Sections 9.3.1 through 9.3.7.1) to detail Software Safety Activities in paragraph/table format to clearly display inputs, tasks, and outputs.
- Section 5.0 is the Software Verification and Validation Plan (SVVP) (Revision 2, Section 7.0)
  - Revised Subsection 5.3.1 through 5.3.8.11 (Revision 2, Section 7.3.1 through 7.3.8.11) to display V&V Activities in paragraph/table format to clearly display inputs, tasks, and outputs.
- Section 6.0 is the Software Configuration Management Plan (SCMP) (Revision 2, Section, 10.0)
- Section 7.0 is the Software Test Plan, (Revision 2, Section 8.0, “Tests”),
  - The following new sections were added to this plan.
    - 7.1 Purpose
    - 7.2 Scope
- Section 8.0 is SQAPM Maintenance (Revision 2, Section 18.0)

### **Content Changes**

- Renamed the Software Quality Assurance Plan (SQAP) to the Software Quality Assurance Program Manual (SQAPM). This update is reflected throughout the contents of this Manual.
- Reference updated to reflect the renaming of the Software Management Plan (SMP) to the Software Management Program Manual (SMPM). This update is reflected throughout the contents of this Manual.
- Revised document reference in new Subsection 3.3 to read, “This SMP [2.3(1)]” to “This SQAP [2.3(1)].
- Reference to “This SVVP Outlines” to “This SQAPM SVVP Outlines” in new Subsection 5.1.2
- Added Item #8 to new Subsection 5.4.7 to read “8. A summary to describe the basis for the selected Software Integrity Level (SIL)”
- Revised the Overview Subsection, 1.1 to read,

“The SQAPM describes the Software Quality Assurance (SQA) activities to be performed during the software life cycle phases of the ESBWR for Nuclear Safety-Related (Quality Class Q) and Nonsafety-Related (Quality Class N3 and N2) digital computer-based I&C system, hereafter referred to as software product.

The SQAPM meets the acceptance criteria specified in Chapter 7 of NUREG 0800, Standard Review Plan (SRP) [2.2.1(1)], except where specified in Appendix A.

In addition, the SQAPM meets the requirements specified in the ESBWR Man-Machine Interface System and Human Factors Engineering Implementation Plan (MMIS/HFE IP) [2.1(1)] for an ESBWR - I&C Software Quality Assurance Program Manual (SQAPM) to be prepared.”

- Added additional 5 IEEE’s to Subsection 2.2.4:
  - IEEE-603-1991 including correction sheet dated January 30, 1995 - IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations
  - IEEE 7-4.3.2-2003 IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations
  - 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

- Added additional 5 IEEE Guidance to list in Subsection 2.4:
  - 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 12207-1996 - IEEE/Electronic Industries Alliance (EIA) Standard for Software Life Cycle Processes
- Deleted revision reference in Subsection 2.1 list of supporting documents
- Table 2 Updated
- Table 3 Updated reference throughout to reflect new supplemental documents table.
- Table 4 Updated
- Appendix C Updated
- Appendix B Updated
- Appendix A Updated
- Table in Subsection 2.3 Updated
- Table 1.5-1, Software Classification, Updated
- Figure 4 updated (including formatting)
- Revised paragraph 3 in new Subsection 5.3.1 to read, “For software V&V, traceability shall be established to show the linkage between design input requirements and the implementation of these requirements throughout the software lifecycle processes. The following subsection describe the Management of V&V tasks,”
- Added additional sentence in new Subsection 3.2.3.4.4, Software Safety Team to read, “The SST is also responsible for evaluating the modifiability and style characteristics of the Software Requirements Specification (SRS).”
- Revised new Subsection 5.3.1 through 5.3.8.11 (Rev 2 Sections 7.3.1 through 7.3.8.11) to detail V&V Activities in paragraph/table format to clarify on inputs, tasks, and outputs.
- Revised last paragraph of new Subsection 5.4.5 to read, “The BRR is forwarded to the responsible Technical Project Engineer (TPE) to be filed in the project DRF. The responsible TPE has the overall technical responsibility for assuring that the hardware and software design of a software product meets the specified requirements.”

- Revised first sentence of Subsection 1.5 to read, “Classification shall be conducted on all software products to assign the appropriate Software Classification as described in the Table 1.5-1.”
- Revised GE Energy Nuclear (GEEN) to read GE Hitachi Nuclear Energy (GEH) throughout.
- Corrected Subsection 6.7.1 reference to Software Change Management Plan (SCMP) to read, “Software Configuration Management Plan (SCMP).
- Removed “Analysis” from the title in Subsection 4.3.3.1. Title now reads, Design Tasks and Documents.
- Deleted the word “phase” from Subsection 6.1.1 and 6.2.2 (1a).
- Revised Subsection 3.9.1, Vendor Control with the addition to “manager” in sentence, “The IVVT shall support the SQA Manager during vendor audit.”
- Corrected new Section 6.0, Record Collection and Retention run on sentence to read, “The baselined configuration items stored on a magnetic or optical medium shall undergo periodic archival backup in accordance with GEH Quality Record Computer Data [2.3(2.s)]. This document prescribes the requirements, procedures, and responsibilities for the control, retention, and retrieval of quality-related computer-based data maintained within the central computing facility of GEH.”
- EOP reference changes throughout document
- Addressed grammatical and editorial issues throughout, such as capitalization consistency, misspellings/typos, missing punctuations, etc.

**Contents**

1. Introduction.....	1
1.1 Overview.....	1
1.2 Purpose and Scope.....	1
1.3 Acronyms, Abbreviations and Definitions.....	2
1.4 Software Developed by Vendors.....	2
1.5 Software Classification.....	2
2. Applicable Documents.....	6
2.1 Supporting Documents.....	6
2.2 Codes and Standards.....	6
2.2.1 NUREG.....	6
2.2.2 Code of Federal Regulations.....	6
2.2.3 U.S. Nuclear Regulatory Commission Regulatory Guides.....	7
2.2.4 Institute of Electrical and Electronics Engineers.....	7
2.3 Supplemental Documents.....	9
2.4 Additional IEEE Standard Guidance.....	14
2.5 International Standards.....	14
3. Software Quality Assurance Plan.....	15
3.1 Purpose and Scope.....	15
3.2 Management Organization.....	15
3.2.1 Organization.....	15
3.2.2 Activities.....	16
3.2.3 Qualification and Responsibilities.....	16
3.2.4 Organizational Interfaces.....	22
3.2.5 Scheduling and Planning.....	23
3.2.6 Approval Authority.....	23
3.3 Documentation.....	24
3.4 Standards, Practices, Conventions and Metrics.....	24
3.4.1 Standards, Practices and Conventions.....	24
3.4.2 Metrics.....	24
3.5 Reviews and Audits.....	25
3.5.1 Reviews.....	25
3.5.2 Audits.....	26
3.6 Problem Reporting and Corrective Action.....	27
3.6.1 Problem Reporting.....	27
3.6.2 Corrective Action.....	27
3.7 Tools, Techniques and Methodologies.....	28
3.7.1 Tools.....	28
3.7.2 Techniques and Methodologies.....	31
3.8 Code and Media Control.....	31
3.9 Vendor and Acquired Software Control.....	31
3.9.1 Vendor Control.....	31
3.9.2 Commercial Off-the-Shelf Software.....	32
3.9.3 Previously Developed Software.....	32
3.10 Records Collection, Maintenance, and Retention.....	33
3.11 Training.....	33

3.12 Risk Management .....	33
4. Software Safety Plan.....	34
4.1 Purpose and Scope .....	34
4.2 Software Safety Management .....	34
4.2.1 Organization and Responsibilities .....	34
4.2.2 Qualifications and Training .....	35
4.2.3 Software Life Cycle .....	35
4.2.4 Documentation Requirements.....	35
4.2.5 Software Safety Program Records .....	36
4.2.6 Software Configuration Management Activities .....	37
4.2.7 Software Quality Assurance Activities .....	37
4.2.8 Software Verification and Validation Activities.....	37
4.2.9 Tool Support and Approval.....	37
4.2.10 Previously Developed or Purchased Software .....	38
4.2.11 Subcontract Management.....	38
4.2.12 Process Certification .....	38
4.3 Software Safety Analyses .....	38
4.3.1 Software Safety Analyses Preparation .....	40
4.3.2 Software Safety Requirements Analysis.....	42
4.3.3 [[ ]]	44
4.3.4 [[ ]]	46
4.3.5 [[ ]]	48
4.3.6 [[ ]]	49
4.3.7 [[ ]]	50
4.4 Post Development .....	51
4.4.1 Training.....	51
4.4.2 Deployment.....	51
4.4.3 Monitoring .....	52
4.4.4 Maintenance.....	52
4.4.5 Retirement and Notification.....	52
4.4.6 Plan Approval .....	52
4.5 Software Safety Analysis Report .....	52
5. Software Verification and Validation Plan.....	54
5.1 Purpose and Scope .....	54
5.1.1 Purpose.....	54
5.1.2 Scope.....	54
5.2 V&V Overview.....	55
5.2.1 Organization.....	55
5.2.2 V&V Schedule .....	55
5.2.3 Software Integrity Level Scheme.....	55
5.2.4 Resources Summary.....	55
5.2.5 Roles and Responsibilities .....	55
5.2.6 Tools, Techniques, and Methods .....	56
5.3 V&V Activities and Tasks .....	59
5.3.1 [[ ]]	60
5.3.2 [[ ]]	62
5.3.3 [[ ]]	67
5.3.4 [[ ]]	71
5.3.5 [[ ]]	76

5.3.6	[[ ]]	83
5.3.7	[[ ]]	86
5.3.8	[[ ]]	92
5.3.9	Acquired Software and Vendor V&V Tasks	95
5.4	V&V Reporting	96
5.4.1	[[ ]]	97
5.4.2	[[ ]]	97
5.4.3	[[ ]]	97
5.4.4	[[ ]]	97
5.4.5	[[ ]]	98
5.4.6	[[ ]]	98
5.4.7	[[ ]]	99
5.5	[[ ]]	99
5.5.1	[[ ]]	100
5.5.2	[[ ]]	100
5.5.3	[[ ]]	101
5.5.4	[[ ]]	101
5.5.5	[[ ]]	101
5.5.6	[[ ]]	101
6.	Software Configuration Management Plan	102
6.1	Purpose and Scope	102
6.1.1	Purpose	102
6.1.2	Scope	102
6.2	Software Configuration Management	103
6.2.1	Organization	103
6.2.2	SCM Responsibilities	103
6.2.3	Applicable Policies, Procedures, and Directives	106
6.2.4	SCM Schedule	106
6.3	Software Configuration Management Resources	106
6.3.1	SCM Tools	106
6.3.2	SCM Techniques	107
6.4	SCM Tasks	107
6.4.1	Configuration Identification	107
6.4.2	Configuration Control	108
6.4.3	Configuration Status Accounting	114
6.4.4	Configuration Audits	114
6.4.5	Baseline Reviews	115
6.5	Software Release Procedures	117
6.6	Software Product Release	117
6.7	Vendor Control	118
6.7.1	Software Developed by Vendors for the Project	118
6.7.2	Acquired Software	118
6.8	Record Collection and Retention	119
7.	Software Test Plan	120
7.1	Purpose	120
7.2	Scope	120
7.2.1	[[ ]]	120
7.2.2	[[ ]]	120
7.2.3	Test Submittal	122

---

7.3 [[	]] .....	123
7.4 [[	]] .....	123
7.5 [[	]] .....	124
7.6 [[	]] .....	124
7.7 [[	]] .....	125
7.7.1 [[	]] .....	126
7.7.2 [[	]] .....	127
7.7.3 [[	]] .....	128
8. SQAPM Maintenance .....		130
9. Tables & Figures .....		131

**List of Tables**

Table 1.5-1, Software Classification.....	3
Table 6.4.2.2-1, Reasons for Change Request.....	110
Table 6.4.2.2-2, Change Process Steps.....	111
Table 1-1, Software Life Cycle Tasks, Responsibilities and Documentation-Planning Phase ..	132
Table 1-2, Requirements Phase.....	133
Table 1-3, Design Phase .....	134
Table 1-4, Implementation Phase .....	136
Table 1-5, Test Phase.....	138
Table 1-6, Installation Phase.....	139
Table 1-7, Operations and Maintenance Phase.....	142
Table 2, V&V and SSA Tasks Assigned to Each Software Class.....	143
Table 3, Problems and Corrective Action Reporting.....	150
Table 4, Configuration Items .....	153

**List of Figures**

[[	]]	.....	4
[[	]]	.....	17
Figure 3, Example of a Traceability Matrix Structure.....			29
Figure 4, Baseline Review Record.....			156
Figure 5, Software Library Structure.....			157

## **1. INTRODUCTION**

### **1.1 OVERVIEW**

The SQAPM describes the Software Quality Assurance (SQA) activities to be performed during the software life cycle phases of the ESBWR for Nuclear Safety-Related (Quality Class Q) and Nonsafety-Related (Quality Class N3 and N2) digital computer-based I&C system, hereafter referred to as "software product."

The SQAPM meets the acceptance criteria specified in Chapter 7 of NUREG 0800, Standard Review Plan (SRP) [2.2.1(1)] and Branch Technical Position (BTP) HICB-14 R4, Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems, except where specified in Appendix A.

In addition, the SQAPM meets the requirements specified in the ESBWR Man-Machine Interface System and Human Factors Engineering Implementation Plan (MMIS/HFE IP) [2.1(1)] for an ESBWR - I&C Software Quality Assurance Program Manual (SQAPM) to be prepared.

### **1.2 PURPOSE AND SCOPE**

The purpose of the SQAPM is to:

- Establish an SQA program in full compliance with 10 CFR 50, Appendix A, General Design Criteria for Nuclear Power Plants and IEEE 603 IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations.
- Monitor the software life cycle activities of the software products and to identify the organization responsible for the SQA program and its organizational boundaries.
- Supplement the GE Hitachi Nuclear Energy (GEH) Quality Assurance Program, which is in full compliance with 10 CFR 50, Appendix B, Quality Assurance Criteria for Nuclear Power Plant and Fuel Processing Plants [2.2.2].

The objectives of the SQA program are to ensure that:

- The design teams comply with:
  - Regulatory compliant GEH Policies and Procedures (P&Ps) to guide software development
  - The Engineering Operating Procedures (EOPs)
  - The requirements described in this SQAPM
  - The ESBWR I&C Software Management Program Manual (SMPM) [2.3(1)] (hereafter referred to as SMPM)

- The design documentation and design outputs for each software life cycle phase defined in the SMPM [2.3(1)] are adequate (i.e., correct and complete).
- The final software products are high quality, acceptable for installation, and ready for reliable operation in a nuclear power plant.

The SQAPM defines the SQA activities, methods, and tools necessary to execute these objectives. The SQAPM also specifies the following:

- Required Verification and Validation (V&V) activities [Section 5.0, Software V&V Plan (SVVP)]
- Software Safety Analysis (SSA) [(Section 4.0 Software Safety Plan (SSP)]
- Software Configuration Management (SCM) [Section 6.0 Software Configuration Management Plan (SCMP)]
- Software Testing Requirements [Section 7.0 Software Test Plan (STP)]

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

The applicable Software Products (software and firmware) covered by this SQAPM encompass all ESBWR I&C systems, as specifically defined in the MMIS/HFE IP [2.1(1)] (Subsection 1.2.4 only), which perform the monitoring, control, 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 are provided in Appendix C.

### **1.4 SOFTWARE DEVELOPED BY VENDORS**

Software Products developed by GE vendors shall comply with this SQAPM. If a vendor elects to follow its established SQA program, then the SQA program as defined in the contract/purchase order (Subsection 3.9, Vendor and Acquired Software Control) shall be reviewed and approved by the SQA Manager to assure compliance with the requirements specified in this SQAPM.

### **1.5 SOFTWARE CLASSIFICATION**

All software products shall be assigned the appropriate Software Classification as described in Table 1.5-1.

If the software performs functions, which are classified per the GEH Safety-Related Classification determination process [2.3(2.t)], then it shall be classified as Software Class "Q."

All other software shall be considered nonsafety-related and will be divided into two sub-classes “N3” and “N2.” A criticality analysis shall be conducted for all nonsafety-related software. If there is a failure mode, which could challenge safety-related systems, then the software shall be classified as “N3.”

The remaining Software shall be classified as “N2.” This software is nonsafety-related system software whose failure cannot adversely affect a safety-related function.

The Software Classification is determined as shown in Figure 1 and is performed during the SSA Preparation phase as described in Subsection 4.3.1. This scheme is based on IEEE Std. 1012, IEEE Standard for Verification and Validation Plans [2.2.4].

**Table 1.5-1, Software Classification**

Classification	Description
Software Class Q	Software performs functions classified per GEH Safety-Related Classification determination process [2.3(2.t)] as safety-related.
Software Class N3	<p>Nonsafety-related systems software whose failure could challenge safety-related systems as defined below:</p> <ul style="list-style-type: none"> <li>• Software whose inadvertent response to stimuli, failure to respond when required, response out-of-sequence could result in an accident or transient as defined in the Design Control Document, Chapter 15 [2.1(6)]</li> <li>• Software that is intended to mitigate the result of an accident</li> <li>• Software that is intended to support recovery from the result of an accident</li> </ul>
Software Class N2	<ul style="list-style-type: none"> <li>• Software failure cannot adversely affect a safety-related function</li> <li>• Software failure results in inconvenience to the user</li> </ul>

[[

]]

[[

]]

[[

]]

## **2. APPLICABLE DOCUMENTS**

Applicable documents include supporting documents, codes and standards, and supplemental documents. Supporting documents provide the input requirements to this plan. Supplemental documents are used in conjunction with this plan.

### **2.1 SUPPORTING DOCUMENTS**

The following supporting documents were used as the controlling input documents in the production of this program. These documents form the design basis for the activities stated in this plan. This document governs, in the event of any differences noted between the SQAPM and the ESBWR Composite Design Specification.

- 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 CODES AND STANDARDS**

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

#### **2.2.1 NUREG**

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

- 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**

- 10 CFR 50, Appendix - B, Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants

### **2.2.3 U.S. Nuclear Regulatory Commission Regulatory Guides**

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

- 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
- RG 1.152-2006, Criteria for Use of Computers in Safety Systems of Nuclear Power Plants

### **2.2.4 Institute of Electrical and Electronics Engineers**

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

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

- IEEE 7-4.3.2-2003, IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations
- IEEE 603-1991 including correction sheet dated January 30, 1995, IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations
- IEEE 828-1990, Standard for Software Configuration Management Plans

- IEEE 829-1983, Standard for Software Test Documentation
- IEEE 830-1993, IEEE Recommended Practice for Software Requirements Specifications
- IEEE 1008-1987, IEEE Standard for Software Unit Testing
- IEEE 1012-1998, Standard for Software Verification and Validation
- IEEE 1028-1997, Standard for Software Reviews
- IEEE 1042-1987, Guide to Software Configuration Management
- IEEE 1074-1995, IEEE Standard for Developing Software Life Cycle Processes

### 2.3 SUPPLEMENTAL DOCUMENTS

The following supplemental documents are used in conjunction with the SQAPM and enable the performance of the activities stated in Appendix A.

Reference Number	Document Title	Document Number
1.	ESBWR Software Management Program Manual (SMPM)	NEDO-33226
<b>GE-Hitachi Nuclear Energy Procedures and Policies</b>		
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	Product Data Management System (PDMS)	PDMS is the computer-based data system that stores, retrieves, and reports data relevant to the engineering definition of products and services offered and provided to customers. It provides current listings of the engineering documents under formal GEH change control (i.e., engineering controlled documents) that have been approved for issue or application to specific standard, requisition, fuel, and operating plant projects.
2.c	Supplier Design Services Document Review	Defines responsibilities and procedural requirements for review, approval, and control of documentation from suppliers for design services. Supplier submitted documents are entered as elements of the design basis in the Product Data Management System as engineering controlled documents or Design Record Files.
2.d	Engineering Test	Defines the process for specifying, performing, evaluating, and documenting engineering tests.
2.e	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.f	Design Process	Defines the process for performing, documenting, and certifying design activities. Design activities include developing or modifying the design of systems, hardware and software, and the performance or modification of licensing studies, engineering evaluations, analyses, calculations and document preparation (e.g., specifications, drawings, reports).
2.g	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.h	Material Requests	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 GEH contract/purchase order, code, and regulatory requirements.
2.i	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.j	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.k	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 ensures traceability, configuration, and quality assurance of engineering documents are maintained through the current document revision, status, and final disposition.
2.l	Procurement Initiation and Control	Specifies the requirements for procurement of material, equipment, and services, including the application of technical, engineering, customer, and quality requirements to purchase orders. Defines the requirements for establishing and maintaining the Approved Suppliers List.
2.m	Supplier Supporting Document Review	Defines responsibilities and procedural requirements for review and acceptance of supporting documents submitted by suppliers of material, equipment and services to satisfy GEH Purchase Order requirements.
2.n	Deviation Disposition Requests from Suppliers	Establishes the requirements and procedure for processing Deviation Disposition Requests submitted by suppliers to obtain a disposition of deviations from the technical requirements of GEH Purchase Order requirements.

2.o	Supplier Change Request	Defines supplier responsibility and procedural requirements for the submittal of a Supplier Change Request to obtain an exception or change to GEH Purchase Order requirements.
2.p	Engineering Change Control	<p>Establishes the process used to control and authorize changes to engineering controlled documents to:</p> <ul style="list-style-type: none"> <li>a. Assure that total impact is considered before a change is approved and that the affected documents are identified and changed as approved</li> <li>b. Provide authority for a change and identify all pertinent interfaces and organizations responsible for these interfaces</li> <li>c. Provide accurate and traceable records of a change</li> </ul>
2.q	Field Deviation Disposition Request	Establishes a process to document and disposition the technical position for field deviations to 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.r	Change Control Board	Defines the requirements and procedures applicable to the operation of a Change Control Board that is responsible for reviewing proposed changes to design or product configuration documents. Establishment of a Change Control Board and application of this procedure are at the discretion of project management for any particular project or group of projects.
2.s	Quality Record Computer Data	Prescribes the requirements, procedures, and responsibilities for the control, retention, and retrieval of quality record computer-based data maintained within the central computing facility of GEH. It includes, but is not restricted to, textual data, computer databases, computer program source data, and binary computer programs.

2.t	Safety-Related Classification	<p>Defines the requirements used to identify structures, systems, components, parts, and technical services that are safety-related.</p> <p>Safety-related structures, systems, components, and parts provide safety-related functions necessary to assure:</p> <ul style="list-style-type: none"> <li>a. The integrity of the reactor coolant pressure boundary; or</li> <li>b. The capability to shut down the reactor and maintain it in a safe shutdown condition; or</li> <li>c. The capability to prevent or mitigate the consequences of accidents that could result in potential off site exposures comparable to 10 CFR 50.34(a)(1) or 10 CFR 100.11 guideline exposures, as applicable.</li> </ul>
2.u	Dedication of Commercial Grade Items	<p>Establishes the requirements and responsibilities for dedicating commercial grade items procured for use in safety-related applications.</p>
2.v	Self Assessment, Corrective Action and Audits	<p>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 GEH.</p>
2.w	Control of Nonconforming Material	<p>Describes the methods by which nonconforming material is documented and controlled at GEH.</p>
2.x	Quality and Technical Training	<p>Defines the roles and responsibilities to assure personnel proficiency in quality and technical related activities. The Quality and Technical Training program:</p> <ul style="list-style-type: none"> <li>a. Ensures personnel are trained and proficient in assigned quality and technical tasks.</li> <li>b. Documents qualifications for technical positions, including minimum education, experience, and any special training requirements.</li> <li>c. Records training assignments in a centralized controlled training database.</li> </ul>

3.a	Project Risk Management Procedure	Implements the project risk management requirements of GEH Policy. Provides a controlled process for risk management to maintain positive control of work situations, especially during critical tasks or activities.
3.b	Project Management Policy	Provides requirements for the single Project Management process across all GEH. The process components include project initiation, planning, scheduling, execution, controls, and post-delivery closeout.
3.c	Quality Policy and Quality System Requirements	Establishes the requirements of the 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.d	Nuclear Energy Quality Assurance Audit Requirements	Establishes the requirements and processes for a comprehensive audit program to verify the implementation and effectiveness of the 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.e	Reporting of Defects and Noncompliance Under 10 CFR Part 21	Defines the requirements and responsibilities within GEH for ensuring compliance with the requirements of 10 CFR 21, "Reporting of Defects and Noncompliance."
<b>Reference Number</b>	<b>Document Title</b>	<b>Document Number</b>
4.	IEEE Standard Glossary of Software Engineering Terminology	IEEE 610.12-1990
5.	Guidelines on Evaluation and Acceptance of Commercial Grade Digital Equipment in Nuclear Safety Applications	EPRI TR-106439
6.	ESBWR Cyber Security Program Plan	NEDO-33295

## **2.4 ADDITIONAL IEEE STANDARD GUIDANCE**

The following IEEE Standards provide additional guidance for the implementation activities. Conformance of this plan to these activities has been evaluated. Selected sections/topics from these IEEE Standards are excluded from commitment because either they provide conflicting requirements with other Standards or the level of detail is not appropriate for this plan. 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/Electronic Industries Alliance (EIA) Standard for Software Life Cycle Processes

## **2.5 INTERNATIONAL STANDARDS**

- International Standards Organization (ISO) 9001:2000, Quality Management Systems - Requirements

### **3. SOFTWARE QUALITY ASSURANCE PLAN**

#### **3.1 PURPOSE AND SCOPE**

The purpose of this Software Quality Assurance Plan (SQAP) is to define the management organization, techniques, procedures, and methodologies used to assure the delivery of software which meets specified requirements for the ESBWR I&C Systems. The use of this plan will help assure the following:

- That software development, evaluation and acceptance standards, are implemented, documented, and followed.
- That the results of software quality reviews and audits will be given to appropriate management within the scope of the SQAPM. This provides feedback as to how the development effort is conforming to development standards.
- That test results adhere to acceptance standards.

#### **3.2 MANAGEMENT ORGANIZATION**

##### **3.2.1 Organization**

This section defines the functional responsibilities and authorities of the ESBWR Project organizations that are responsible for the quality of the software products. The organization of the ESBWR Project is shown in Figure 2.

The Quality organization is responsible for GEH Quality Assurance (QA) program. The Quality Organization is a managerially and financially independent organization. The Quality Manager, who reports to the President and CEO of GEH, provides leadership for development and overall coordination of the QA program objectives, including the software quality assurance program. The SQA organization has the overall responsibility for developing and maintaining the SQA program with support from the Software Project Engineering (SPE) organization. The SPE organization is responsible for executing the technical aspects of the SQA program, which includes the following SQA tasks (hereafter referred to as Quality tasks):

- Independent Verification and Validation (IV&V) of Software Class Q software
- Software Safety Analysis (SSA)
- Software Configuration Management (SCM)

The SPE organization is technically, managerially, and financially independent from the software products design organization, in conformance with RG 1.168 [2.2.3].

### 3.2.2 Activities

The following activities are performed throughout the software life cycle phases:

- Verification and Validation of design documentation and outputs specified in the SMPM [2.3(1)]
- Safety Analysis of Software Class Q software and Software Class N3 software
- Software and System Testing
- Baseline Reviews
- Configuration Control
- Audits

Table 2 outlines the tasks and the individual or group responsible for conducting these tasks during the design and development of the software products.

### 3.2.3 Qualification and Responsibilities

The SQA Manager shall be knowledgeable in the industry standard QA methodologies and proficient in establishing, maintaining, and improving QA Procedures and experience in technical project management. The SPE members shall be knowledgeable in the technologies and methods used in design development and are qualified to perform the specific software quality assurance tasks.

[[

]]

The level of software quality assurance support varies during each software life cycle phase; thus, the membership of the SQA and SPE fluctuates with the level of needs. If necessary, the SQA and SPE Manager have the authority to contract third party organizations (e.g., consultants or experts in I&C software design and development for nuclear power plants) to support the software quality assurance activities.

[[

[[

]]

]]

### ***3.2.3.1 Software Quality Assurance Manager***

The SQA Manager, who interfaces with the SPE Manager, has the overall responsibility and authority of the SQA Program. The SQA Manager is responsible for:

- Approving this SQAPM
- Approving or rejecting the validated software
- Issuing stop work order if the audit or assessment findings indicate violation of the quality and/or safety requirements
- Organizing the software auditing activities and maintaining the software audit plan
- Participating in baseline reviews
- Scheduling and coordinating software audits (both internally and externally) with the New Plant Project (NPP) Quality Team and/or the Nuclear Quality Assurance Team to ensure effectiveness of the audit being conducted
- Reporting audit results to the responsible project leadership (e.g., SPE Manager, Engineering Manager, Project Management Team) and the Quality Manager

### ***3.2.3.2 New Plant Project Quality Manager***

The New Plant Project (NPP) Quality Manager has the overall responsibility and authority of the Quality Program for the ESBWR Project. The NPP Quality Manager shall coordinate with the SQA Manager concerning the audit of the software products. The NPP Quality Manager is responsible for:

- Quality assurance requirements for the design and production of the software products. This includes but is not limited to:
  - Hardware production
  - Hardware qualification
  - Shipping and packaging
  - Final product quality certification
  - Release for shipping approval
- Organization of the auditing activities and maintenance of the audit plan

The NPP Quality Manager shall either reserve authority, or shall formally designate a Quality Control Engineer (QCE) who has the authority, to reject or order stop work when such action is deemed necessary to assure the quality or safety of the software product.

The NPP Quality Manager is also responsible for assuring that adequate resources are available to support the QA program. Quality initiatives for improvement of processes used by GEH for product and service offerings include:

- Reporting to top management on the performance of the quality assurance system
- Ensuring the promotion of awareness of quality requirements throughout the organization

### ***3.2.3.3 I&C and Electrical Systems Engineering Manager***

The I&C and Electrical Systems Engineering organization, hereafter referred to as the Design Team, is described in the SMPM [2.3(1)]. The I&C and Electrical Systems Engineering Manager has the overall responsibility to ensure the design and development of the software products are performed in accordance with the SMPM [2.3(1)] and the required GEH procedures and policies. This includes the approval or rejection of the design documentation, timely and effective control of work in process, and the quality of delivered software products.

### ***3.2.3.4 Software Project Engineering***

The SPE is independent of the design team to ensure organizational freedom to perform the Quality tasks without undue pressure or conflict of interest related to budget and schedule. The following SPE teams are responsible for executing the Quality tasks described in this plan (Subsection 3.2.2). A Task Lead is assigned by the SPE Manager to lead each of the following SPE teams:

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

#### ***3.2.3.4.1 SPE Manager***

The SPE Manager has the overall responsibility and authority for the implementation of the Quality tasks during the software life cycle. The SPE Manager is responsible for:

- Coordinating with the SQA Manager in organizing the Quality tasks
- Approving the IV&V, SSA, or baseline review design outputs and documentation
- Rejecting the design outputs and documentation, as recommended by the Task Lead(s), if serious defects are identified during SSA and/or IV&V (e.g., the requirements and/or design are incomplete, inconsistent, and not traceable to upper level documents)
- Staffing of BRT, SST, and IVVT
- Appointing the Task Lead to BRT, SST, and IVVT

- Communicating open issues to Engineering organizations and the Project Management Team, and Quality Manager
- The overall management, including schedule and budget of SPE, to ensure continued effectiveness and support of the Quality tasks described in this SQAPM

#### **3.2.3.4.2 Independent Verification & Validation Team**

The IVVT is responsible for performing and managing IV&V tasks on the Software Class Q design documentation and design outputs to:

- Ensure the design meets the specified requirements
- Confirm the quality, safety, reliability, availability, maintainability, testability, security, and performance of the design
- Ensure the software products meet their intended use and do not perform unintended functions

The IVVT Task Lead is responsible for:

- Organizing the IV&V tasks and coordinating the IV&V schedule with the Design Team
- Assigning IV&V tasks to IVVT Team members
- Managing the conduct of IV&V tasks, and reviewing and approving the IV&V reports prepared by IVVT Team members
- Ensuring that the IV&V is performed in accordance with the SVVP described in Section 5.0

#### **3.2.3.4.3 Baseline Review Team**

The BRT is responsible for performing the baseline review to assess the adequacy of the software design process and control of configuration items (CIs) in accordance with the Software Configuration Management Plan (SCMP) (Section 6.0). The BRT shall issue a Baseline Review Record (BRR) for each baseline review conducted.

The BRT Task Lead is responsible for:

- Coordinating and scheduling the baseline review meetings with the Design Team and the SQA Manager
- Organizing the baseline review process
- Assigning tasks to BRT members
- Managing the conduct of BRT tasks

- Ensuring that baseline review tasks and activities are performed in accordance with the SCMP described in Section 6.0
- Approving the baseline configuration items
- Coordinating the release of configuration items into the Configuration Management System (CMS)

#### **3.2.3.4.4 Software Safety Team**

The Software Safety Team (SST) is responsible for performing the SSA to assure the safety characteristics of the software products being developed, including the interface between the hardware and software. The SST has the authority to enforce safety requirements in the software requirements specification (SRS), the design, and the implementation of the software. The SST is also responsible for evaluating the modifiability and style characteristics of the Software Requirements Specification (SRS).

The SST is responsible for determining the Software Class (described in Subsection 1.5, Software Classification) of the software and performing SSA as appropriate. The SST shall coordinate with the IVVT to evaluate the verification efforts to determine if SSA can be used as a verification method.

The SST Task Lead is responsible for:

- Overseeing the overall conduct of the software safety program
- Organizing the software safety program and coordinating the SSA schedule with the IVVT and the Design Team
- Approving or rejecting safety software
- Assigning tasks to SST Team members
- Managing the conduct of SST tasks and approving SSA reports prepared by SST members
- Ensuring that each SSA is performed in accordance with the SSP described in Section 4.0
- Establishing supplemental tasks to support SSA

### **3.2.3.5 Configuration Management Manager**

The Configuration Management Manager (CMM) has the overall responsibility and authority for the CMS. Configuration Management is responsible for defining the CM process and tools, as well as execution of the CMS to maintain and control traceable records of:

- Design Requirements and Inputs
- Design Activities
- Design Output
- Authorizations to execute change requests to the controlled records
- Approvals of the execution of change requests

### **3.2.3.6 Design Team**

The Design Team is responsible for the design and implementation of the software products and the independent verification and validation of Software Class N3 and N2 software products. The responsible verifiers and testers shall be individual(s) or groups(s) who are competent to perform verification and validation based on knowledge and experience. The V&V shall be conducted by individuals(s) or groups(s) other than those who performed the design of the software product.

The independence criteria are defined in GEH Independent Design Verification [2.3(2.i)].

The roles and responsibilities of the Design Team are described in the SMPM [2.3(1)].

The Design Team is responsible for the independent verification of software class N3 and N2 software, per Independent Design Verification [2.3(2.i)].

### **3.2.4 Organizational Interfaces**

Figure 2, SPE and Quality Organizational Functions Interfaces, depicts the interfaces between the SPE and the I&C and Electrical Systems Engineering, NPP, Quality, and vendors.

The NPP Project Managers (PMs) are responsible for the commercial aspects of the software project, including:

- Management and control of vendors' participation in the design and delivery of the software products
- Maintaining coordination between SPE and vendor organizations
- Vendors' performance; PMs have the authority to request an audit of any vendor
- Interface with customer (i.e. owner/user), hereafter referred to as "Licensee"

---

The detailed responsibility of the PM is described in the SMPM [2.3(1)]. The SQA Manager, with support of the NPP Quality Team, shall perform SQA audits on the external vendor organizations prior to contract agreement (Subsection 3.9.1, Vendor Control). Vendors responsible for producing software products within the scope of the SQAPM shall be in compliance with the requirements specified in this SQAPM and the regulatory requirements described in the SMPM [2.3(1)].

### **3.2.5 Scheduling and Planning**

The SPE and SQA Managers have the overall responsibility for scheduling and planning the tasks and activities describe in this SQAPM. The Task Lead for each team (SST, IVVT, BRT) is responsible for the management and planning activities for their respective teams. The Task Leads shall coordinate with the Design Teams concerning the timely receipt of design documentation to support the quality tasks (SSA, IV&V, baseline review [BR], and software audit).

Schedule and project planning shall be documented in a Project Work Plan (PWP) in accordance with GEH Work Planning and Scheduling [2.3(2.a)]. Each Task Lead is responsible for the preparation of a task specific PWP.

While the Quality tasks are performed by a cross-functional team, a project workflow shall be established to ensure the required tasks are accurately identified and the Quality tasks schedule is aligned with the established integrated project schedule and milestones. The schedule shall:

- Cover the duration of the SQAPM
- Contain the major milestones of the project related to the Quality tasks
- Include the sequence and dependencies of the Quality tasks and the relationship of key Quality tasks to project milestones
- Express milestones as absolute dates

### **3.2.6 Approval Authority**

The NPP Quality Manager, the SQA Manager, and the SPE Manager have approval authority for functions within their responsibility.

Upon the rejection of a software product or the issuance of a stop work order, corrective actions shall be established that may include a correction or amendment of the design process, revision to the software plans, re-design, re-implementation, or re-testing of the software product. The Design Team shall be required to complete the corrective actions and identify preventive actions to avert the occurrence of similar defects.

### 3.3 DOCUMENTATION

The SMPM [2.3(1)] establishes the managerial process and the technical direction necessary to govern the design and development activities of the software products. The required design documents and design outputs to be prepared are defined in the SMPM [2.3(1)].

[[

]]

### 3.4 STANDARDS, PRACTICES, CONVENTIONS AND METRICS

#### 3.4.1 Standards, Practices and Conventions

The applicable GEH EOPs and P&Ps used in guiding the design and development of software products are specified in Subsection 2.3, Supplemental Documents. If detailed instructions are needed, project or platform/product line specific work practice instructions, such as ESBWR Project Instructions (EPI), Engineering Service Instructions (ESI), or Work Instructions (WI) are prepared to provide additional instructions as required. Software audits shall be conducted to monitor compliance to the policies and procedures used to guide the design and development of software products.

Software coding shall be implemented in accordance with the guidelines defined in the Software Coding Conventions and Guidelines document required by the SMPM [2.3(1)], which at a minimum, shall include:

- Documentation standards
- Logic structure standards
- Coding standards
- Commentary standards

Code review shall validate coding compliance to the guidelines outlined in the (applicable) Software Coding Conventions and Guidelines document.

#### 3.4.2 Metrics

Software Metrics are sets of data that are systematically collected and analyzed to provide software quality process feedback to the software development processes. This feedback mechanism provides a means by which the software development processes can change over time to facilitate continuous process improvement with the primary objective of producing high quality defect free software products. Specific metrics will be defined for each software platform or product line and for each software classification.

The metrics program shall focus on the software functional and process characteristics listed in Appendix D. These characteristics will be used to derive a core set of metrics relating to the

development process and the design documentation and outputs, such as requirements and design documents, code, and test documentation.

The SPE will be responsible for collecting and analyzing metric data for the software Class Q and N3 software products.

### **3.5 REVIEWS AND AUDITS**

#### **3.5.1 Reviews**

##### ***3.5.1.1 Technical Review***

The purpose of the technical review is for a qualified individual, or a team of qualified individuals, to determine the suitability of the intended use of a design and identify discrepancies from design inputs, codes, and standards. It ensures the following:

- The design document conforms to its specifications
- The design document adheres to regulations, standards, guidelines, plans, and procedures applicable to the project
- The design document is complete and correct
- For a document in revision, the changes have been implemented as specified in the change request or anomaly report

Technical review may be conducted through peer review or design review.

Peer review shall be conducted by an individual other than the design document's Responsible Engineer (RE). A Peer review is considered an informal review and cannot be used to replace independent verification. The RE shall document and disposition the review comments. The review comments shall be filed in the project Design Record File (DRF).

[[

]]

##### ***3.5.1.2 Managerial Review***

The SPE Manager, the SQA Manager, and the Task Leads shall review the SQA program in accordance with Procedure GEH Quality Policy and Quality System Requirements [2.3(3.c)], to ensure its suitability, adequacy, and effectiveness. The review team shall assess opportunities for improvement and the need for changes to the SQA program and quality objectives. In addition to the review inputs required by Procedure GEH Quality Policy and Quality System Requirements [2.3(3.c)], the review includes the effectiveness of V&V, SSA, and SCM tasks to monitor compliance with the defined requirements.

The review shall be documented in the Managerial Review Report [5.4.6] that shall include decisions and actions needed to assure continued effectiveness of the SQA program. Maintenance of the SQAPM is described in Section 8.0, SQAPM Maintenance.

### ***3.5.1.3 Project Closeout Review***

The responsible PM shall schedule a post-delivery closeout review to formally terminate the activities of a project, such as closing any Corrective Action Requests (CARs) associated with the project and project Design Record File (DRF), setting up warranty administration and review, and conducting a Licensee closeout meeting to solicit feedback, which includes collecting lessons learned and metrics during the project.

The post delivery closeout review shall be conducted in accordance with GEH Project Management Policy [2.3(3.b)].

## **3.5.2 Audits**

### ***3.5.2.1 Functional Audit***

The functional audits shall be conducted to assure that the requirements specified in the System Design Specification (SDS) and Software Requirements Specification (SRS) have been met by checking the applicable Requirements Traceability Matrix (RTM). The functional audit shall be performed during baseline review by the BRT and shall be documented in the Baseline Review Record. The functional audit shall be performed for the Software Class Q software products and is recommended for Software Class N3 and N2 software products.

### ***3.5.2.2 Physical Audit***

The physical audits shall be conducted to verify that the appropriate CI item, which include Software Build Description (SBD), has accurately and completely described the "build" parameters of the software such that a duplicate version of the object code can be recreated. The physical audit shall be performed as part of Test Phase Baseline Review by the BRT and shall be documented in the Test Phase Baseline Review Record. The physical audit shall be performed for Software Class Q software products and is recommended for Software Class N3 and N2 software products.

### ***3.5.2.3 In-Process Audits***

This SQAPM requires SQA audits to be performed on the design organizations (both internal and external) that are working on the ESBWR Project. The SQA audit shall be performed to ensure compliance with the codes and standards specified in this SQAPM. The SQA audit evaluates the adequacy and completeness of the required reviews and V&V activities.

GEH Nuclear Energy Quality Assurance Audit Requirements [2.3(3.d)], establishes the requirements and processes for a comprehensive audit program to confirm implementation of

and compliance with the GEH quality system and to determine the adequacy and effectiveness of the quality system.

An audit report shall be prepared at the conclusion of each software audit. The audit report shall summarize the following:

- Audit activities and results
- Audit observations
- Conditions Adverse to Quality (CAQs)
- Discrepancies
- Non-compliance with quality and engineering procedures
- Recommended corrective actions

A CAR shall be initiated to manage the identified CAQs, discrepancies, and non-compliances in accordance with the procedure specified in GEH Self Assessment, Corrective Action and Audits, [(2.3(2.v))].

### **3.6 PROBLEM REPORTING AND CORRECTIVE ACTION**

#### **3.6.1 Problem Reporting**

Discrepancies, deficiencies, anomalies, deviations or comments discovered during design and development (i.e. V&V, SSA and testing), installation, post delivery, and other CAQs shall be formally documented. Table 3 outlines the problem reporting process, including possible scenarios, responsible individuals, and documentation of reported problems.

Defects and non-compliance under 10 CFR Part 21 shall be reported in accordance with GEH Reporting of Defects and non-compliance under 10 CFR Part 21 [2.3(3.e)].

#### **3.6.2 Corrective Action**

It is essential that the process requirements described in this SQAPM, the SMPM [2.3(1)]; the required EOPs, P&Ps, and Corporate QA program be adhered to. Failure to comply with these requirements shall be promptly identified and action shall be taken to eliminate or correct the nonconformities or CAQs to prevent recurrence. CAQs can be:

- Discovered during work performance and audit
- A complaint from licensees
- Findings from regulatory authorities
- Other external organizations (e.g., International Standards Organization (ISO)/American Society of Mechanical Engineers (ASME) Code authorities)

---

GEH Self Assessment, Corrective Action and Audits [2.3(2.v)] defines the procedures and responsibilities for reporting and recording nonconformance or CAQ. A CAR is used to document a nonconformity or CAQ or an opportunity for process/product improvement. A CAR is also used to ensure timely evaluation is performed and to record objective evidence of actions taken to correct the CAQ and preventive action(s) to prevent future occurrence. The Commitment Tracking System (CTS) is used to manage and record the identified CAQ (Subsection 3.7.1.1).

## **3.7 TOOLS, TECHNIQUES AND METHODOLOGIES**

### **3.7.1 Tools**

The SPE organization shall employ the use of tools as needed to execute the tasks specified in this plan. Tools used in part to perform V&V tasks do not need to be qualified if V&V is performed on the output produced by the tool.

#### ***3.7.1.1 Commitment Tracking System***

The Commitment Tracking System (CTS) is used to manage and record the identified CAQs and non-compliances to the established quality procedures, such as EOPs, P&Ps, and this SQAPM as defined in Section 2.0.

#### ***3.7.1.2 Checklist***

A checklist may be used to support inspection, independent verification and software audits to ensure completeness of the design output being verified or inspected, and the process being audited. The checklists prepared to support software inspection and independent verification should include acceptance criteria for the design output. The NUREG 0800, SRP [2.2.1(1)] divides the acceptance criteria into two sets:

- Functional characteristics (accuracy, functionality, reliability, robustness, safety, security, or timing). Not all characteristics occur for every design output.
- Process characteristics (completeness, consistency, correctness, style, traceability, unambiguity, or verifiability). Not all characteristics occur for every design output.

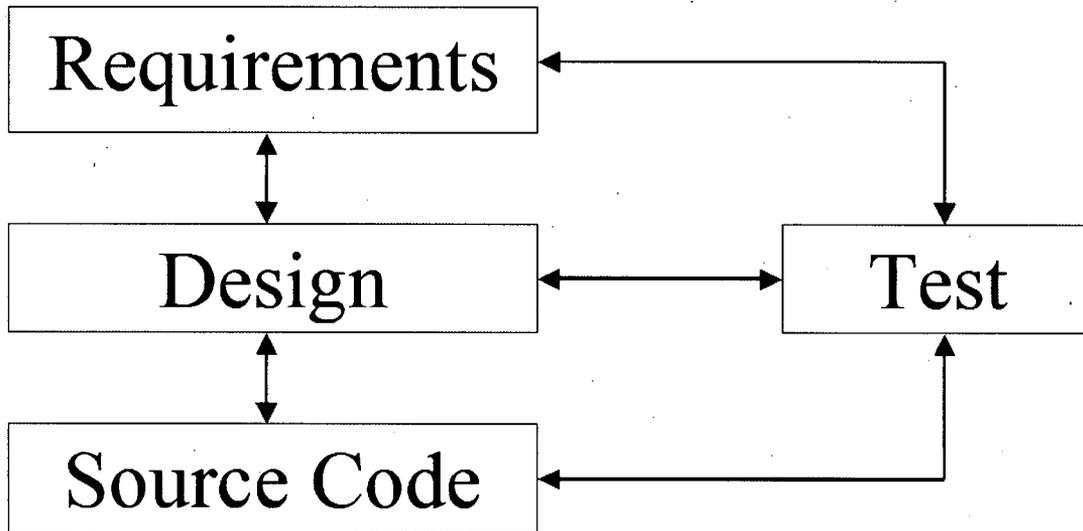
Software audits are conducted to independently evaluate the design team's compliance with the SQA requirements specified in this SQAPM and other applicable standards, regulations, guidelines, and procedures. The checklists prepared to support software audits should include queries to demonstrate compliance with the SQA requirements specified in this SQAPM and other applicable standards, regulations, guidelines, and procedures.

#### ***3.7.1.3 Requirements Traceability Matrix***

The Design team is responsible for the preparation of Requirements Traceability Matrix (RTM). RTM can be prepared manually or using an automated tool.

[[

]]



**Figure 3, Example of a Traceability Matrix Structure**

#### ***3.7.1.4 Product Data Management System***

The GEH Product Data Management System (PDMS) [2.3(2.b)] is an access controlled, computer-based data storage and retrieval system that is used to manage data relevant to the engineering definition of products and services, including quality records. Previous and current revisions of the engineering documents that have been approved for issue are maintained with the system. Roles, responsibilities and procedures are defined within PDMS.

PDMS is the GEH official CMS for engineering and quality controlled documents. Internal and external vendors providing the ESBWR software products are not required to utilize PDMS. However, an appropriate computer-based CMS shall be used.

### **3.7.1.5 Design Record File**

A Design Record File (DRF) is the GEH formal controlled information record for engineering work. DRF records include the following:

- Design of systems, hardware and software
- Performance of analyses, evaluations, calculations
- Documentation from licensing services

A DRF is an in-progress record, subject to change, that is contained in the PDMS until it becomes a permanent Quality Assurance Record. A DRF shall be generated and maintained in accordance with GEH Design Record File, [2.3(2.g)].

### **3.7.1.6 Discrepancy Tracking System**

A Discrepancy Tracking System will be initiated to manage and track the anomalies identified during software testing. [[

]] Data that will allow the RE to evaluate the anomaly shall be included in the Discrepancy Tracking System. At a minimum, the following information shall be included:

- Name and Master Parts List (MPL) of the software product
- Project / Plant
- Software Classification
- Description of the anomaly, including effect and extent of the anomaly, clear explanation of observations, symptoms, workarounds, and any other pertinent information
- Severity of the Anomaly
- Initiation Date
- Affected Documentation
- Affected Design Organization
- Corrective Action/Resolution Statement
- Completion and Approval Status

If needed, reports can be generated to:

- Facilitate and monitor the anomaly disposition efforts

- Ensure that the required changes to the affected design documentation and output have been completed
- Support baseline review and management review efforts

### **3.7.2 Techniques and Methodologies**

Techniques and methodologies used to support the Quality tasks are described in the SVVP (Section 5.0), SSP (Section 4.0), and SCMP (Section 6.0).

### **3.8 CODE AND MEDIA CONTROL**

The computer-based design outputs, such as software source code, Commercial Off-the-Self (COTS) software, and support software/tools used to support the design and development of software products are CIs; and, as such, shall be controlled as specified in the SCMP (Section 6.0).

### **3.9 VENDOR AND ACQUIRED SOFTWARE CONTROL**

#### **3.9.1 Vendor Control**

Vendor selection and qualification shall be performed under a prescribed process. As a minimum, the following requirements shall be evaluated:

- Ability to meet engineering, quality, and purchasing requirements
- Relevant experience in the design and development of similar products
- Awareness of and compliance with the applicable regulatory and industrial requirements
- Service, installation, and support capability and history of performance

Confirmation of this ability is determined by audits and/or reviews of the vendor's Quality Management System, including the Quality Assurance Program. The IVVT shall support the SQA Manager during vendor audit.

[[

]]

### **3.9.2 Commercial Off-the-Shelf Software**

Commercial Off-the-Shelf (COTS) software is software commercially available to anyone. COTS software includes communication protocol applications and linkable software libraries. It is acceptable that the qualified and dedicated COTS software be used in the Software Class Q software products. The SMPM [2.3(1)] describes the qualification and dedication of COTS software.

### **3.9.3 Previously Developed Software**

Previously Developed Software (PDS) is software developed for prior projects and not necessarily verified and validated per the requirements outlined in this SVVP. The IVVT shall independently verify the PDS evaluation report prepared by the design team for software intended for use in Class Q systems. The SMPM [2.3(1)] describes the evaluation and dedication of PDS.

### **3.10 RECORDS COLLECTION, MAINTENANCE, AND RETENTION**

Subsection 6.8, Record Collection and Retention describes the collection, maintenance, and retention of design documentation, design outputs, and quality records, such as audit reports, SSA reports, and test reports.

### **3.11 TRAINING**

All personnel supporting the Quality tasks shall be trained, as necessary, to ensure proficiency in applicable quality and technical tasks prior to the assignment of work activities affecting the quality of software products as required by GEH Quality and Technical Training [2.3(2.x)]. The design team and the SPE teams shall be trained, either by self-study or classroom, in this SQAPM, the SMPM [2.3(1)], applicable tools required to support the design and V&V tasks, and the referenced EOPs and P&Ps. The training records shall be maintained in appropriate ESBWR and GEH training databases.

### **3.12 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.a)].

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.

## **4. SOFTWARE SAFETY PLAN**

### **4.1 PURPOSE AND SCOPE**

This Software Safety Plan (SSP) establishes the processes and activities intended to ensure that the safety concerns of the software products are properly considered during the software development and are consistent with the defined system safety analyses as defined by RG 1.173, Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants [2.2.3(6)]. This SSP meets the guidelines specified in Chapter 7 of NUREG 0800 SRP [2.2.1(1)] and the requirements outlined in Section 4.4 of IEEE Std. 1228, "IEEE Standard for Software Safety Plans" [2.4(1)].

Safety is the most important consideration, taking precedence over budget and schedule.

Software Safety Analysis (SSA) is performed on the software for Software Class Q and N3 software products.

### **4.2 SOFTWARE SAFETY MANAGEMENT**

#### **4.2.1 Organization and Responsibilities**

SQAPM Subsection 3.2.1 describes the organization efforts in supporting the SSA activities. The roles and responsibilities of the Software Safety Team (SST) are described in SQAPM Subsection 3.2.3.4.4.

The SPE Manager and the SST Task Lead have responsibility and authority for the completion of SSA activities:

- Prepare the SSA plan
- Obtain and allocate resources to ensure effective implementation of the Plan including budgeting, qualified personnel, and suitable training or continuing education to keep personnel current
- Coordinate safety task planning with other organizational components or functions, such as development, system safety, software quality assurance, software reliability, software configuration management, V&V, and software testing
- Coordinate software safety tasks within the overall context of the system safety program
- Coordinate technical issues related to software safety with other components of the development and support organization, with the project sponsor, or with the customer
- Ensure that adequate records are kept to document the conduct of software safety activities
- Participate in audits of software safety plan implementation

- Ensure training of safety and other appropriate personnel in methods, tools, and techniques used in software safety tasks
- Ensure that any deviations and/or discrepancies are identified, documented and dispositioned prior to proceeding

#### **4.2.2 Qualifications and Training**

Personnel assigned specific responsibilities for SSA shall meet the qualifications and on-going training as defined in SQAPM Subsection 3.2.3.

#### **4.2.3 Software Life Cycle**

The software safety process shall be integrated into the software life cycle process defined in SMPM Subsection 5.3, Organization of Software Life Cycle Process [2.3(1)].

#### **4.2.4 Documentation Requirements**

- Software Project Management. Documentation of how the software safety program will be implemented, integrated, and managed with the software development activities is discussed in this SQAPM and in the SMPM [2.3(1)].
- Software Configuration Management. Information regarding the CM of the design documentation and design outputs is specified in Section 6.0, Software Configuration Management Plan.
- Software Quality Assurance. Information regarding the SQA of software design documentation and design outputs is specified in Section 3.0 of this SQAPM.
- Software Safety Requirements. Specification of safety requirements to be met by the software to avoid or control system hazards will be performed during development of the Software Requirements Specification as specified in the SMPM [2.3(1)].
- Software Safety Design. Descriptions of the software design elements that satisfy the software safety requirements are specified in the SMPM [2.3(1)] Subsection 5.8.3.1, Software Design Description.
- Software development methodology, standards, practices, metrics, and conventions. Approved and controlled practices that are essential to satisfy system and software safety objectives and requirements are specified in the SMPM [2.3(1)] Subsection 5.8.3.3, Software Coding Conventions and Guidelines Document.
- Test Documentation. Software safety test planning, test design, test cases, test procedures, and test reports are specified in Subsection 7.7, Test Documentation.
- Software Verification and Validation. Information regarding how software safety will be verified and validated is defined in Section 5.0, Software V&V Plan. The software safety analyses are specified in this SSP. The Requirements Traceability

Analysis (RTA) (Subsection 5.2.6.2.6) is used to ensure the traceability of safety requirements to the design specifications, software source code, and software safety test cases.

- Reporting Safety Verification and Validation. Information documenting the results of software Safety-Related Verification and Validation activities is specified in Subsection 5.4, Verification and Validation Reporting.
- Software User Documentation. Information that may be significant to the safe installation, use, maintenance, and/or retirement of the software product is specified in the SMPM [2.3(1)] 8.0, Software Operations and Maintenance Plan.
- Results of Software Safety Requirements Analysis. The reporting requirements for this activity are specified in Subsection 4.3.2, Software Safety Requirements Analysis.
- Results of Software Safety Design Analysis. The reporting requirements for this activity are specified in Subsection 4.3.3, Software Safety Design Analysis.
- Results of Software Safety Code Evaluation. The reporting requirements for this activity, including software functional testing, are specified in Subsection 4.3.4, Software Safety Code Analysis.
- Results of Software Safety Test Analysis. The reporting requirements for this activity are specified in Subsection 4.3.5, Software Safety Test Analysis.
- Results of Software Safety Change Analysis. The reporting requirements for this activity are specified in Subsection 4.3.7, Software Safety Change Analysis.
- Results of Deviations and/or Discrepancies. Deviations and/or discrepancies identified during performance of software safety activities shall be documented and dispositioned prior to proceeding in accordance with the Software Quality Assurance Plan described in Subsection 3.6 of this SQAPM.

#### **4.2.5 Software Safety Program Records**

Records of software safety program activities shall be generated and maintained in accordance with SQAPM Section 3.10, Records Collection, Maintenance and Retention. Records addressing the following topics shall be maintained.

- Results of analyses, including V&V, performed on requirements, design, code, test, and other technical documentation
- Information on suspected or confirmed safety problems in the pre-release or installed system
- Results of audits performed on software safety program tasks
- Results of safety tests conducted on all or any part of the entire system

- A record of training provided to software safety program personnel
- Results of any certifications performed
- Results of deviations and/or discrepancies

#### **4.2.6 Software Configuration Management Activities**

Software Configuration Management shall be in force during all phases of the software life cycle and shall be accomplished in accordance with SQAPM Section 6.0, Software Configuration Management Plan.

#### **4.2.7 Software Quality Assurance Activities**

Software Quality Assurance activities, as described in this SQAPM, shall assure proper performance of key software safety program activities. Software Quality Assurance activities shall include, at a minimum:

- The software safety plan is prepared, approved, implemented, changed, and made consistent with predecessor document.
- The technical recommendations resulting from software safety tasks are reviewed, considered by change control authority, and, where appropriate, implemented.
- The reviews and audits will address software safety concerns, requirements, guidelines, and process certification.
- The conduct of the software safety program will be monitored.

#### **4.2.8 Software Verification and Validation Activities**

Software Safety Verification and Validation (V&V) activities are specified in SQAPM Section 5.0, Software Verification and Validation Plan. Verification and validation activities shall ensure that:

- All system safety requirements have been satisfied by the life cycle phases
- No additional hazards have been introduced by the work done during the life cycle activity

#### **4.2.9 Tool Support and Approval**

Software tools used in the development and evaluation of software class Q and N3 software shall be evaluated for suitability. Software tools used to aid the development or evaluation of the software are managed as specified in the SMPM [2.3(1)]. Configuration control of software tools is managed in accordance with the requirements of the SCMP (Section 6.0).

To lessen the possibility of inadvertent introduction of software hazards by project tools, the following areas shall be addressed:

- Tool approval for use on the project
- Installation of upgrades to previously approved tools
- Withdrawal of a previously approved tool
- Identification of limitation that may be imposed on tool use

#### **4.2.10 Previously Developed or Purchased Software**

The SST has the authority to reject the use of the PDS and COTS software in Software Class Q and Class N3 software if the PDS and COTS software does not meet the requirements of this plan.

#### **4.2.11 Subcontract Management**

Management of subcontractors for safety-related software shall be carried out in accordance with the requirements of the SVVP (Section 5.0). Any software activities performed by a subcontractor will either be performed in accordance with this plan or an alternate plan approved by GEH before the performance of the activity.

The Responsible System Engineer (RSE) shall be responsible for ensuring that hazards impacting or identified by the subcontractor are communicated to any affected organizations.

#### **4.2.12 Process Certification**

Process certification, performed to certify that the software product is produced in accordance with the software safety assessment, is achieved through baseline reviews. Baseline reviews are described in Subsection 5.2.6.2.5.

### **4.3 SOFTWARE SAFETY ANALYSES**

The Software Safety Analysis (SSA) is performed to ensure that:

- The system safety requirements have been correctly addressed by the life cycle phases
- No additional hazards have been introduced by the work done during the life cycle activity
- Software elements that can affect safety are identified
- There is evidence that other software and system elements do not affect safety
- Safety problems and resolutions identified in these analyses are documented

The SSA shall be conducted to determine the Software Class as described in Subsection 1.5, Software Class. The SSA shall focus on the software identified as Software Class Q and N3. The process for conducting an SSA is shown in the Verification Activity Table, Subsection 4.3.1

#### 4.3.1 Software Safety Analyses Preparation

- The purpose of the SSA is to establish the fundamental nuclear power plant design characteristics as they relate to the design and implementation of the safety-related software used in plant systems. Generically, this is the identification of all software systems and classifications that should be designated as safety-related.
- As system identification and classification have been performed regardless of the use of the software, the SST can use these as the analysis preparation of Preliminary Hazard Analysis (PHA). At this phase, safety systems may or may not be implemented using software. Safety systems that the design organization identifies as having no software in their implementation are outside the scope of this plan.

[[

]]

Software Quality Assurance Program Manual	Inputs	Tasks	Outputs
4.3.1.1 II			
4.3.1.2			II

Software Quality Assurance Program Manual	Inputs	Tasks	Outputs

### 4.3.2 Software Safety Requirements Analysis

- A preliminary hazard analysis is performed at the requirements phase of the software life cycle to evaluate potential errors and deficiencies in the requirements that may contribute to a hazard.

[[

]]

Software Quality Assurance Program Manual	Inputs	Tasks	Outputs
4.3.2.1			

4.3.3 [[

]]

[[

]]

Software Quality Assurance Program Manual	Inputs	Tasks	Outputs
4.3.3.1 ]]			]]

4.3.4 [[

]]

[[

]]

Software Quality Assurance Program Manual	Inputs	Tasks	Outputs
4.3.4.1 []			[]

4.3.5 [[ ]]

[[

]]

Software Quality Assurance Program Manual	Inputs	Tasks	Outputs
4.3.5.1 [[			]]

4.3.6 [[  
 ]]

]]

Software Quality Assurance Program Manual	Inputs	Tasks	Outputs
4.3.6.1 [[			]]

4.3.7 [[ ]]

[[

]]

Software Quality Assurance Program Manual	Inputs	Tasks	Outputs
4.3.7.1 //			]]

## **4.4 POST DEVELOPMENT**

### **4.4.1 Training**

The Software Training Plan (STRngP) is described in Section 9.0 of the SMPM [2.3(1)].

### **4.4.2 Deployment**

#### ***4.4.2.1 Installation***

Installation is described in Section 7.0 of the SMPM [2.3(1)]. Installation V&V tasks are described in Subsection 5.3.7 and the SAT is described in Section 7.6 of this SQAPM.

#### ***4.4.2.2 Startup and Transition***

Prior to starting up the newly installed software product, the discrepancy log shall be reviewed and evaluated, pre-operational tests shall be conducted to demonstrate the installed software product operates as intended; and, if applicable, the required set points (e.g., trip and alarm) will be established. The pre-operational test shall be conducted in accordance with an approved (by the Licensee) test plan and procedure.

The pre-operational test is outside the scope of this SQAPM as it is usually the Licensee's responsibility and shall be supported by qualified engineers who are knowledgeable in the installed software product and plant operation.

The Startup Procedure shall address the requirements for safely starting the new system and, if an old system is to be replaced, for making a safe transition from the old system to the new system. At a minimum, the following shall be addressed:

- Fallback modes for the new system
- Startup of backup components and subsystems
- Startup of the new system
- Parallel operation with backups
- Parallel operation of the old system and the new system
- Subsystem vs. full system operation
- Switchover to full system operation
- Validation of results from the new system
- Cross validation of results between the old system and the new system

- Fallback in the case of failure of the new system, including fallback to an old system if one exists

#### **4.4.2.3 Operations Support**

The Software Operation and Maintenance (O&M) Manual and User Interface Specification (UIS) shall be provided for the software safety modules. The Software O&M Manual and User Interface Specification are described in Section 8.0 and Subsection 5.7.8 of the SMPM [2.3(1)], respectively.

#### **4.4.3 Monitoring**

[[

]]

#### **4.4.4 Maintenance**

Software maintenance is specified in the Software O&M Manual. The Software O&M Manual is described in Subsection 8.5.1 of the SMPM [2.3(1)].

#### **4.4.5 Retirement and Notification**

Retirement and notification are described in Subsection 5.13 of the SMPM [2.3(1)].

#### **4.4.6 Plan Approval**

Design review of this plan will be performed in accordance with GEH Design Review [2.3(2.e)].

The review basis for this plan is the ESBWR MMIS/HFE IP [2.1(1)] and RG 1.173 [2.2.3(6)]. The SPE manager shall have authority to approve this plan upon completion of the design review.

### **4.5 SOFTWARE SAFETY ANALYSIS REPORT**

As a minimum, the software safety analysis shall include the following:

- Name and Description of the Software Evaluated
- System
- Software Classification
- Purpose and Scope

- Reference Inputs
- Software Safety Analysis Body of Report
- Anomalies Noted
- Conclusion
- Responsible Engineer
- Approving Authority

The report shall be placed under the configuration control as described in Section 6.0, Software Configuration Management Plan.

## **5. SOFTWARE VERIFICATION AND VALIDATION PLAN**

This Software Verification and Validation Plan (SVVP) establishes the V&V tasks for the software designed and developed for software products. This SVVP satisfies the requirements of RG 1.168 [2.2.3], except where specified in Appendix A. RG 1.168 endorses IEEE Std. 1012, IEEE Standard for Verification and Validation Plans [2.2.4] and IEEE Std. 1028, IEEE Standard for Software Reviews and Audits [2.2.4].

### **5.1 PURPOSE AND SCOPE**

#### **5.1.1 Purpose**

The purpose of this SVVP is to outline the specific V&V steps required during the software development process to ensure that:

- The developed software meets its specified requirements, performs its intended functions correctly, and does not perform any unintended function.
- The final software product meets the contract requirements, required industry and regulatory standards, and licensing commitments.
- The final software product is correct, complete, accurate, and traceable to requirements specified in the design documents and outputs.

The goal of this SVVP is to assure that software V&V activities are integrated throughout the software life cycle to facilitate the timely detection of errors and to ensure the quality of the software product.

#### **5.1.2 Scope**

This SQAPM SVVP outlines the formal set of standards and processes necessary to comprehensively verify and validate Class Q, N3, and N2 software products during all phases of the software life cycle. The software life cycle phases in the SQAPM correspond with those defined in the SMPM [2.3(1)].

The V&V tasks, covered by this scope, are described in Subsection 5.3. Software V&V activities shall also be included to analyze and test the software with respect to its hardware interfaces and user interactions. V&V activity is limited to software prepared by GEH and GEH vendors for the ESBWR project and evaluation of the qualification results of Commercial Off-the-Shelf (COTS) software to be used in a Software Class Q software product. Qualification of COTS software is performed by the Design Team as described in Subsection 5.8.3.6 of the SMPM [2.3(1)].

## **5.2 V&V OVERVIEW**

### **5.2.1 Organization**

Subsection 3.2, Management Organization, describes the organization efforts in supporting the Verification and Validation (V&V) activities.

### **5.2.2 V&V Schedule**

The V&V schedule and contingency planning to identify risks shall be documented in the IV&V Tasks PWP (Software Class Q) and the project PWP (Software Class N3 and N2).

### **5.2.3 Software Integrity Level Scheme**

This SQAPM uses an approach similar to the software integrity level scheme specified in IEEE 1012 [2.2.4] to define the V&V requirements for the software product. IEEE 1012 does not mandate the use of the software integrity scheme specified in the standard. The approach used by this SQAPM is described in Subsection 1.5, Software Classification.

### **5.2.4 Resources Summary**

Subsection 3.2.3.4.2, Independent Verification and Validation Team (IVVT), describes the personnel required to support IV&V activities for Software Class Q software. The SPE Manager is responsible for IVVT staffing and budget. The Design Team is responsible for the Software Class N3 and N2 software V&V.

Subsection 3.2.3.4.3, Baseline Review Team (BRT), describes the personnel required to support the baseline review activities.

Subsection 5.2.6, Tools, Techniques, and Methods, addresses the tools, techniques, and methods used to support the V&V activities.

### **5.2.5 Roles and Responsibilities**

The roles and responsibilities of IVVT members are described in Subsection 3.2.3.4.2, Independent Verification and Validation Team. Subsection 3.2.3.1, SQA Manager, describes Quality Organization support in the V&V activities.

The Design Team is responsible for the Software Class N3 and N2 software V&V. The roles and responsibilities of the Design Team are described in the SMPM [2.3(1)]. For Software Class Q software, the Responsible Technical Project Engineer (RTPE) shall formally notify the IVVT Task Lead via a formal project letter when a design document is ready for IV&V.

The Responsible Technical Project Engineer (RTPE) shall formally notify the BRT Task Lead via a formal project letter when a software life cycle phase is ready for baseline review.

The project letters shall be filed in the project DRF.

Table 2 lists the V&V tasks and the individual or group responsible for performing these tasks.

## **5.2.6 Tools, Techniques, and Methods**

### **5.2.6.1 *V&V Tools***

Tools used to support the V&V tasks shall be evaluated. The evaluation results shall be documented in the tool evaluation report.

[[

]]

### **5.2.6.2 *Techniques and Methods***

#### **5.2.6.2.1 Verification**

Verification is performed to determine whether or not the design document/output for a given software life cycle phase fulfilled (i.e., is traceable to) the requirements established in the previous phase.

Verification is also performed to determine if the design documentation and design outputs are complete, consistent, and correct, and will support the next phase.

[[

]]

#### **5.2.6.2.2 Code Review**

Code reviews are performed to verify that the software correctly implements the specified design and does so in a manner that is compliant with the guidelines outlined in the applicable Software Conventions and Guidelines document. Code review shall be performed by a qualified software engineer other than the individual responsible for code implementation.

[[

]]

#### **5.2.6.2.3 Software Functional Test**

The software functional test includes the software module/unit test and the software integration test.

[[

]]

#### **5.2.6.2.4 Software Validation Test**

The software validation test is performed to validate that the software product is operational and conforms to the functional and performance requirements specified in the System Design Specification (SDS), Hardware/Software Specification (HSS), and Software Requirements Specification (SRS).

[[

]]

#### **5.2.6.2.5 Baseline Reviews**

Baseline Reviews are formal, independent evaluations of the software design and development activities performed at the completion of each software life cycle phase.

[[

]]

#### **5.2.6.2.6 Requirements Traceability Analysis**

Requirements Traceability Analysis (RTA) is performed for Software Class Q, N3 and N2 software requirements.

[[

]]

#### **5.2.6.2.7 Audit Support**

Subsection 3.5.2.3 describes the in-process audit.

[[

]]

#### **5.2.6.2.8 Walk-Through**

Design walk-through is a static analysis technique used during the design and development of the software product, which is used to:

- Identify possible design errors
- Identify violation of design requirements, codes, and standards
- Evaluate alternative implementation approaches

[[

]]

### **5.3 V&V ACTIVITIES AND TASKS**

The following sections describe the V&V activities and tasks to be performed for each life cycle phase.

[[

]]

5.3.1 [[ ]]

[[

]]

- Continuous reviews of the V&V effort
- Revision of the SVVP as necessary based upon updated project schedules and development status
- Coordination of the V&V results with the Design Team, the SST, the BRT, and the SQA Manager

[[

]]

]]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.1.1 [[			]]

5.3.1.2 [[

]]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.1.3 //			
			]]

5.3.2 [[ ]]

[[ ]]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.2.1 [[ ]]			
5.3.2.2			]]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs

*5.3.2.3* [[

*5.3.2.4*

]]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.2.5 [[			
5.3.2.6			]]

5.3.2.7 [[

]]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.2.8 [[			
5.3.2.9			]]

5.3.3 [[ ]]

[[ ]]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.3.1 [[ ]]			
5.3.3.2			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.3.3			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.3.4		e.	11

5.3.3.5 ||

||

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.3.6			
5.3.3.7			

5.3.4 [[ ]]

[[ ]]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.4.1 [[ ]]			
5.3.4.2			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.4.3			
5.3.4.4			II

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs

5.3.4.5 []

[]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.4.6 []			
5.3.4.7			[]

5.3.5 [[ ]]

[[ ]]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.5.1 [[ ]]			
5.3.5.2			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
		<ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li><li>•</li></ul>	

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
		<ul style="list-style-type: none"> <li>•</li> </ul>	
5.3.5.3			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
		<ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li><li>•</li></ul>	

5.3.5.4 [[

]]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.5.5 [[			
5.3.5.6			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
<i>5.3.5.7</i>			
<i>5.3.5.8</i>			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.5.9			
5.3.5.10			]]

5.3.6 [[ ]]

[[ ]]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.6.1 [[ ]]			
5.3.6.2			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.6.3			
5.3.6.4			II

5.3.6.5 [[

]]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.6.6 [[			
5.3.6.7			]]

5.3.7 [[ ]]  
 [[ ]]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.7.1 [[ ]]			
5.3.7.2			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.7.3			
5.3.7.4			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.7.5			
5.3.7.6			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
<i>5.3.7.7</i>			
<i>5.3.7.8</i>			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
<i>5.3.7.9</i>			
<i>5.3.7.10</i>			
<i>5.3.7.11</i>			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs

5.3.8 [[ ]]

[[ ]]

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.8.1 [[ ]]			
5.3.8.2			
5.3.8.3			
5.3.8.4			
5.3.8.5			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
<i>5.3.8.6</i>			
<i>5.3.8.7</i>			
<i>5.3.8.8</i>			

Software Quality Assurance Program Manual	V&V Inputs	V&V Tasks	V&V Outputs
5.3.8.9			11

### 5.3.9 Acquired Software and Vendor V&V Tasks

Acquired software refers to:

- Support software/tools used to support the software development activities.
- COTS software.
- Previously developed software (PDS).

**5.3.9.1** [[

]]

#### **5.3.9.2 COTS Software**

COTS software is software that is commercially available to anyone. It is acceptable that COTS software be used in a Software Class Q application if it is qualified and dedicated in accordance with EPRI TR-106439, Guidelines on Evaluation and Acceptance of Commercial Grade Digital Equipment in Nuclear Safety Applications [2.3(5)]. GEH Dedication of Commercial Grade Items [2.3(2.u)] provides additional guidance in dedicating commercial grade items procured for use in safety-related applications.

[[

**5.3.9.3**

**5.3.9.4**

]]

**5.4 V&V REPORTING**

The report shall be prepared at the conclusion of each V&V task to capture the V&V results and status. The control of the V&V report is described in the SCMP (Section 6.0, Software Configuration Management Plan). Subsection 3.6, Problem Reporting and Corrective Action, describes the V&V process.

5.4.1 [[ ]]

[[

]]

5.4.2 [[ ]]

[[

]]

5.4.3 [[ ]]

[[

]]

5.4.4 [[

]]

5.4.5 [[ ]]

[[

]]

5.4.6 [[ ]]

[[

]]

5.4.7 [[ ]]

[[

]]

5.5 [[ ]]

[[

]]

5.5.1 [[

]]

[[

]]

5.5.2 [[

]]

[[

]]

5.5.3 [[ ]]

[[

]]

5.5.4 [[ ]]

[[

]]

5.5.5 [[ ]]

[[

]]

5.5.6 [[ ]]

[[

]]

## **6. SOFTWARE CONFIGURATION MANAGEMENT PLAN**

### **6.1 PURPOSE AND SCOPE**

This Software Configuration Management Plan (SCMP) establishes the Software Configuration Management (SCM) activities during the design and development of the software products. This SCMP satisfies the requirements of RG 1.169, Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants [2.2.3], except where specified in Appendix A. RG 1.169 endorses IEEE Std. 828, IEEE Standard for SCM Plans [2.2.4]. This SCMP also complies with the planning requirements of IEEE 1042.

#### **6.1.1 Purpose**

The intent of this SCMP is to provide additional guidance and direction necessary to implement the SCM activities required throughout the software product life cycle. This SCMP supplements GEH established configuration management procedures in system and hardware design. It establishes a formal set of standards and methodology used to administer and control the configurations of Software Class Q and Software Class N3 and N2 software products and shall remain in effect throughout the software life cycle.

#### **6.1.2 Scope**

The scope of SCMP includes the following:

- Describes the individual with the overall responsibility and authority for the SCM and organizations responsible for supporting the SCM activities
- Defines the SCM tasks, including methods, timing, and responsibility for the implementation of design control and design change control.
- Identifies the tools, procedures, and individuals needed to execute or support each SCM task.
- Identifies the SCM required schedule and coordination with the design activities and the Quality tasks described in this SQAPM.

## 6.2 SOFTWARE CONFIGURATION MANAGEMENT

### 6.2.1 Organization

The hierarchy of responsibility for the Software Configuration Management (SCM) activities is as follows:

- The Configuration Management Manager (CMM) has the overall responsibility and authority for the CMS, including system maintenance and enhancement.
- The BRT Task Lead has the overall responsibility of the baseline review process and the configuration control of software products.
- The Responsible Configuration Control Engineer (RCCE) is responsible for the configuration control of the design documentation and outputs related to the software product, and the maintenance of software library.
- The BRT is responsible for judging adherence to the software development process for the design documentation and/or outputs being baselined. The members of this team are appointed by the BRT Task Lead and must be independent from the design team responsible for the design documentation and/or outputs.
- The Responsible Manager is responsible for the technical scope (design and development) of the software product.
- The Responsible Engineer is responsible for a given technical item (e.g., the design and development of the documentation).

### 6.2.2 SCM Responsibilities

- The primary responsibilities of the Configuration Management Team, under the direction of the CMM, are to support the following:
  - Design control throughout the software life cycle to ensure compliance with the applicable safety and performance requirements
  - Design change control to establish the change approval criteria for all change requests. (as defined in GEH Change Control Board [2.3(2.r)])
  - Engineering document management to control the project records
  - Engineering document format and issuance to ensure consistency and standardization of the engineering documentation and issuance process are being used and followed
- The Change Control Board (CCB) is responsible for the evaluation of the proposed high impact modifications to the software product design or product configuration

documentation. The CCB also provides recommendations, which include concurrence, rejection, modification, or hold for further investigation.

The requirements and procedures applicable to the operation of the CCB are described in GEH Change Control Board [2.3(2.r)].

High impact modification is a change that affects one or more of the following factors:

- Safety and licensing
- System or plant performance
- Design interface (internal or external)

A detailed list of high impact changes is described in GEH Engineering Change Control [2.3(2.p)].

- The SCM responsibilities of the I&C and Electrical Systems Engineering Manager is to review and approve the initiation or change of design documents for Software Class N3 and N2 software to confirm:
  - Review and verification were performed by technically competent individual(s)
  - Scope of review and verification per GEH Independent Design Verification [2.3(2.i)] is complete
  - Comments made by the reviewers were adequately resolved
- The SCM responsibilities of the responsible SQA manager:
  - Approve/reject the validated software
  - Participate in Baseline Reviews
- The SCM responsibilities of the TPE are to:
  - Identify the reason for the document initiation or change (i.e., error correction, regulatory or Licensee requirement, etc.)
  - Determine the timing of baseline review
  - Identify the items to be baselined
  - Authorize the distribution or release of validated software (i.e., source, object, and executable codes) to RCCE for configuration control
- The SCM responsibilities of the RE are to:
  - Initiate or revise engineering controlled documentation and obtain the required verification documents in accordance with GEH Independent Design Verification [2.3(2.i)]

- Resolve the non-conformances identified
- The responsibilities of the RCCE are to:
  - Ensure the software and associated documentation are entered into the software library after the approval of the BRT
  - Release software source code/application code to the Design Team for revision or the approved software package for production
  - Coordinate software configuration control with Configuration Manager
  - Support baseline review as a BRT member
  - Maintain the software library
- The BRT chairperson is appointed by the BRT Task Lead. The responsibilities of the BRT chairperson are to:
  - Appoint members of the BRT
  - Establish the BRT by assigning review responsibilities
  - Initiate baseline reviews
  - Chair the baseline reviews
  - Document the baseline review meeting, BRT members, and attendees. This information shall be stored in the appropriate DRF
  - Track open baseline items

The BRT members shall have sufficient skill and experience to effectively judge the adequacy of the V&V of the CIs being baselined. The BRT Members shall be knowledgeable in the Baseline Review process. They shall be independent from the design and development of the CIs under review.

The BRT may include individuals from the Quality Organization, the Configuration Management Team, or the Design Team, all of whom are independent of the design and development of the CI subject to baseline. The responsibilities of the BRT are to:

- Ensure that the CIs are properly identified, verified, and controlled
- Ensure compliance with the SMP, SCMP, SVVP, and SSP
- Review and approve the resolved nonconformance comments from baseline reviews

The project manager is responsible for coordinating the release of design documentation to the vendor supporting the project, including the coordination of review and approval by the RE of vendor submittals, the design interface review, and control of project correspondence. Vendor Control is described in Subsection 3.9.1

### **6.2.3 Applicable Policies, Procedures, and Directives**

The Nuclear Energy P&Ps, EOPs, and directives applicable to the SCM activities, are specified in Section 2.0 and its subsections. These policies and procedures are used to supplement the process specified in this SCMP. If any external constraints are placed on the plan per contract requirements, such constraints and its impact and effect on the SCMP shall be documented in the project PWP.

### **6.2.4 SCM Schedule**

The SCM schedule that establishes the sequence and SCM tasks shall be specified in the PWP of the individuals responsible (Subsection 6.2.1) for the SCM tasks. Subsection 3.2.5 describes the scheduling and planning for the Quality tasks, including the tasks related to SCM.

## **6.3 SOFTWARE CONFIGURATION MANAGEMENT RESOURCES**

### **6.3.1 SCM Tools**

The following are the SCM Tools used to support the design of software products:

- Product Data Management System (PDMS) is the GEH official CMS. It is used for the creation, control, approval, storage, and retrieval of documents or data in electronic media. PDMS is described in Subsection 3.7.1.4, Product Data Management System
- Design Record File (DRF) is a formal controlled information record under the GEH procedures for in-progress and completed engineering work which is retained and from which work can be retrieved. DRF is described in Subsection 3.7.1.5, Design Record File
- Human Factors Engineering Issue Tracking System (MMIS/HFE IP [2.1(1)]) is a Web-based database used to track:
  - HFE issues that are not resolved through the normal HFE process
  - Software problems, defects, or anomalies discovered during design and development (not part of V&V activities)
  - Baseline review open items
- Commitment Tracking System (Subsection 3.7.1.1) is used to track:
  - Requirement violations, deviations, repeat procedural violations, and non-conformances
  - Post delivery software and documentation errors and discrepancies
  - Issues identified that are outside the scope of the Design Review, as defined in GEH Design Review [2.3(2.e)]

- Discrepancy Tracking System (Subsection 3.7.1.6) is used to track anomalies identified during software validation test, [[ ]] and applicable, SAT

### **6.3.2 SCM Techniques**

Subsection 6.4 identifies the SCM Tasks, the techniques and procedures used to accomplish each task, the individuals responsible for each task, and applicable tools used to support each task.

## **6.4 SCM TASKS**

### **6.4.1 Configuration Identification**

The CIs subject to this plan include:

- Engineering documents prepared to document the design, to communicate system requirements for software products and material, and to support the implementation and manufacturing of the software products. Engineering documents are typically issued documents. They shall be assigned a unique document identification number, revision status, quality classification, and pagination, including the total number of pages in the document.

GEH Document Initiation or Change by Engineering Review Memorandum/Engineering Change Notice [2.3(2.k)] establishes the requirements for the initiation or change of engineering documents by use of the Engineering Review Memorandum/Engineering Change Notice (ERM/ECN).

- Quality records such as V&V reports (Subsection 5.4), Test Reports (Subsection 7.7.3), Audit Reports (Subsection 3.5.2) and SSA Reports (Criticality Analysis Reports and Hazard Analysis Reports described in Subsections 4.3.1 to 4.3.7) are prepared to document evidence of the quality of CI and/or execution of Quality tasks. These records shall be filed in the project specific Design Record File (DRF). A DRF is the formal controlled information record used to document design activities and retain/protect completed engineering work.

Each DRF shall be assigned a unique identification number. GEH Design Record File [2.3(2.g)] defines the procedures to establish and maintain a DRF.

- Acquired software such as support software/tools, COTS software, and PDS, shall be assigned a unique identification and revision number in accordance with the format described in project level documentation.
- Software, such as source code listings, objects, and executable files shall be assigned a unique file name and revision number.

Each software module/unit source files shall contain a header comment section, which as a minimum, shall include the quality classification and a revision status.

Upon completion of software validation testing, the RCCE or other individual assigned this responsibility shall place the validated software package (such as source code, object code, executable code, and associated data files for each released revision) under configuration control in the assigned software library. The software library shall serve as the final control point and repository for the released computer-based software configuration items. Different software libraries or project-specific software libraries may be used to control the computer-based configuration items as the software products may be implemented using other control product lines or other computer platforms. Procedures shall be established to describe the retrieval and reproduction process of the controlled computer-based software CIs from library storage.

The software library structures shall have a consistent naming convention and an appropriate level of security control (e.g., password control). The security measures implemented shall provide assurance that the integrity of the baselined CI is maintained. Read and write control access to the software library accounts shall be granted to the RCCE. Personnel participating in the design and development of the software product shall only have read access to the software library. Changes to the software libraries can only be made by the Configuration Control Engineer.

Figure 5 shows an example of the naming convention.

- Vendor submittals shall be assigned a unique identification and revision number.

GEH Supplier Supporting Document Review [2.3(2.m)] defines the responsibilities and the procedural requirements for review and acceptance of design documents submitted by vendors.

All CIs shall be placed under configuration control and stored in the PDMS (Subsection 3.7.1.4). Table 4 contains a list of CIs, their structures, retention medium, and life cycle control points.

## **6.4.2 Configuration Control**

### **6.4.2.1 Design Control**

The design of software products is controlled to ensure compliance with the applicable safety and performance requirements. Design control measures are established to achieve the following:

- Definition of design requirements and performance of design activities in a planned, controlled, and orderly manner.
- Specification of appropriate quality requirements and standards in design documents.
- Selection of appropriate design verification methods and implementation by individuals or groups not directly responsible for the original design.

The design process performed shall be composed of the activities necessary for the complete software life cycle. The design process activities include analyses, preparation of specifications and drawings, testing, generation of test reports, and the technical support (i.e., installation and

training) required to complete the design, implementation, installation, operation, and maintenance of the software products.

The design process and required design documentation are described in the SMPM [2.3(1)] and GEH Design Process [2.3(2.f)]. GEH Document Initiation or Change by Engineering Review Memorandum/Engineering Change Notice [2.3(2.k)] describes the configuration control process for initiating new or revised design documentation. The GEH Product Data Management System (PDMS) [2.3(2.b)] is used for this purpose. PDMS is described in Subsection 3.7.1.4. Design verification shall be performed for the design activities. Section 5.0 of the SVVP describes the required V&V tasks to be performed. V&V task outputs shall be retained within the project documentation in each project-specific DRF.

#### ***6.4.2.2 Design Change Control***

The Design Change Control process includes change initiation, review, approval, implementation, disposition, status reporting, document updating, and distribution. The purpose of this process is to:

- Ensure that total impact is considered before a change is approved
- Ensure that the documents are identified and changed after a change is approved
- Provide authority for a change
- Identify pertinent interfaces and organizations responsible for these interfaces
- Provide accurate and traceable records of change
- Ensure a schedule for implementation of approved design changes is established

A change request may be initiated by the Licensee for product enhancement or by anyone observing a problem/error with a software product, as described below. Reasons for a proposed change are categorized in Table 6.4.2.2-1.

Design change that results in modification to the ESBWR certified design shall be processed in accordance with Subsection 3.1.4.2 of the MMIS/HFE IP [2.1(1)].

**Table 6.4.2.2-1, Reasons for Change Request**

Life Cycle Phase	Reasons
Requirements, Design, Implementation	<ul style="list-style-type: none"> <li>• Design requirements</li> <li>• Change in regulatory requirement or codes and standards requirement</li> <li>• Change request from Vendor (Subsection 3.9.1 Vendor Control)</li> </ul>
Tests	<ul style="list-style-type: none"> <li>• Anomaly or error correction during V&amp;V and testing,</li> <li>• Change request from Vendor (Subsection 3.9.1 Vendor Control)</li> </ul>
Installation	<ul style="list-style-type: none"> <li>• Anomaly or error correction during installation</li> </ul>
Operation and Maintenance	<ul style="list-style-type: none"> <li>• Anomaly or error correction during operation and maintenance,</li> <li>• Licensee contract change</li> </ul>

Table 6.4.2.2-2, Change Process Steps

[[

]]

[[	]]
[[	<ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li><li>•</li><li>•</li><li>•</li><li>•</li><li>•</li><li>•</li></ul> ]]
[[	<ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li></ul> ]]

[[	<ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li><li>•</li><li>•</li><li>•</li></ul> ]]
[[	<ul style="list-style-type: none"><li>•</li><li>•</li><li>•</li></ul> ]]

6.4.2.2.1 [[

]]

**6.4.2.2.2 Change Request During V&V and Test Phase**

Subsection 5.5.1 describes the documentation of discrepancies or errors discovered during V&V and testing process. The discrepancies or errors shall be evaluated and resolved. If the discrepancies or errors impact an issued document or multiple documents, the RMCN process or the Engineering Change Authorization (ECA) process described above shall be followed.

**6.4.2.2.3 Change Request During Installation Phase**

Discrepancies or errors discovered in a software product during the installation phase shall be processed using the Field Deviation Disposition Request (FDDR) process per GEH Field Deviation Disposition Request [2.3(2.q)].

**6.4.2.2.4 Change Request During Operations and Maintenance Phase**

Change requests initiated during the Operations and Maintenance Phase as the result of software errors shall be reported and tracked using CTS (Subsection 3.7.1.1).

**6.4.2.2.5 Change Request from Licensee**

Proposed changes to a software product design due to contract revision shall be processed using the Engineering Change Authorization (ECA) process per GEH Engineering Change Control [2.3(2.p)] described above.

**6.4.2.3 Change Notification**

If discrepancies or errors affect products already installed or turned over to the Licensee, the Project Manager shall:

- Notify the affected plant Licensee of any detected non-conformances

- Supply to the affected plant the upgraded software or Erasable Programmable Read-Only Memory (EPROM)

#### **6.4.2.4 Design Interfaces Control**

Engineering design interfaces with vendors or design organizations supporting the design of the software product shall be formally conducted and information formally transmitted. Project correspondence that pertains to the transmission or acceptance of project documents shall be maintained in PDMS.

To ensure interface compatibility, design documents shall be distributed for information and/or review to the affected design organizations to ensure that there is no conflict in the design objectives and to ensure that the product resulting from the interfacing designs function as planned. The PM is responsible for coordinating the distribution of design documents to appropriate design organizations.

[[

]]

#### **6.4.3 Configuration Status Accounting**

Status for design documentation and design outputs can be collected from the PDMS by selecting report module features to obtain the status of the design documentation and design outputs. The responsible TPE shall maintain a record or database used to prepare reports on the status of design documentation and design outputs. The record or database shall include the initial approved version, the status of requested changes, and the implementation status of approved changes for each CI, as well as outstanding engineering documents undergoing engineering change requests that have not yet been resolved. Configuration status reports shall be used as supporting information to the project progress report to ensure timely reporting of project progress and baseline review.

#### **6.4.4 Configuration Audits**

Configuration audits shall be performed on the software CIs (including the computer-based items) to ensure the completeness of the software products. There are two types of configuration audits:

- Functional Configuration Audit
- Physical Configuration Audit

#### **6.4.4.1 Functional Configuration Audit**

A functional configuration audit is performed during the baseline review. The BRT shall inspect the design documentation, outputs, and associated traceability matrix for completeness (i.e., demonstration of forward and backward direction). Deficiencies shall be documented in the functional configuration audit minutes and maintained as an attachment or part of the BRR. The responsible TPE is responsible for ensuring that the deficiencies are corrected.

#### **6.4.4.2 Physical Configuration Audit**

A physical configuration audit is performed during the Test phase baseline review. The BRT shall inspect the Software Build Description of the Software Class Q for completeness, such that a duplicate version of the software package can be recreated. The BRT shall also determine that all items identified as being part of the configuration are present in the product baseline. The audit must establish that the correct version and revision of each part are included in the product baseline and that they correspond to information contained in the baseline's configuration status report. Deficiencies shall be documented in the physical configuration audit minutes and maintained as an attachment or part of the BRR. The responsible TPE is responsible to ensure that the deficiencies are corrected.

#### **6.4.5 Baseline Reviews**

The baseline review is conducted at the completion of each software life cycle phase. For the O&M and Retirement phases, a review of baseline records is performed and revisions are made accordingly. This activity constitutes the baseline review for those phases. The following baselines have been designated by the SMPM [2.3(1)]:

1. Planning
2. Requirements
3. Design
4. Implementation
5. Test
6. Installation
7. Operation and Maintenance
8. Retirement

The SMPM [2.3(1)], in conjunction with the project PWP, specifies the CIs to be baselined during each software life cycle phase.

The purpose of the baseline review is to establish that:

- The design information developed during the software life cycle phase adheres to the software life cycle process outlined in the SMPM
- The V&V tasks and the SSA tasks performed adhere to the procedures outlined in the SVVP and SSP, respectively

The baseline review is performed as follows:

1. Upon completion of the design activities within the software life cycle phase, including the required V&V tasks and SSA, the responsible TPE appoints an engineer to prepare the baseline package. The baseline package consists of CIs to be baselined for the specific software life cycle phase.
2. The responsible TPE shall notify the BRT Task Lead that the design activity for the specific software life cycle phase is completed and is ready for baseline review. The BRT Task Lead shall schedule the baseline review and convene a BRT.
3. The BRT shall be provided with the copies or the depository location of the configuration items (CIs) to be baselined (including the associated V&V reports) prior to the baseline review meeting.
4. A baseline review is performed to assess the design control and design change control, SSA, and V&V tasks of a particular software life-cycle phase.
5. The BRT has the authority to approve or reject the configuration items (CIs) to be baselined. The non-conformances and assessment shall be documented in the BRR (Subsection 5.4.5). The engineer responsible for the baseline package is responsible for resolving these non-conformances. The final resolution of the identified non-conformances shall be documented in the BRR.
6. A baseline review is not complete until the discrepancies have been resolved. However, if the responsible TPE can justify that the discrepancies discovered do not impact the safety and/or security requirements, exception may be granted at the discretion of the BRT to allow the design team to proceed to the next software life cycle phase. This justification must be documented in the BRR or as an attachment to the BRR.
7. The BRT task lead shall prepare the BRR. A copy of the BRR shall be forwarded to the responsible TPE to be filed in the software project DRF.

As software design and development is an iterative process, the baseline review shall be repeated as the baselined configuration item (CI) is modified.

#### ***6.4.5.1 Baseline Items Approval Process***

The configuration items to be baselined must be reviewed by the BRT to confirm that:

- Adherence to the SMPM and SQAPM has been achieved
- The required documents have been completed and verified
- The verification scope and approach is reasonable
- Any comments made during the review process have been adequately documented and that the non-conformances noted have been resolved
- The required testing has been completed, the results documented and verified, and the open issues resolved and approved by the BRT

#### ***6.4.5.2 Baseline Review Record***

The BRT chairperson shall prepare a Baseline Review Record (BRR). Figure 4 provides an acceptable format for the Baseline Review. The BRR is described in Subsection 5.4.5.

### **6.5 SOFTWARE RELEASE PROCEDURES**

The Responsible Configuration Control Engineer (RCCE) has the responsibility and authority for the release of the SQA Manager approved software package for production. The approved software shall be released in accordance with the procedures outlined in the Software Build Description.

### **6.6 SOFTWARE PRODUCT RELEASE**

The responsible project QCE has the authority for the release of the final software product. The Software product is formally released for shipment upon issuance of a Product Quality Certificate (PQC).

## **6.7 VENDOR CONTROL**

Vendor Control is described in Subsection 3.9.1.

### **6.7.1 Software Developed by Vendors for the Project**

The vendor shall utilize this Software Configuration Management Plan (SCMP) to support the design and development of the software products or prepare an equivalent SCMP in accordance with the requirements outlined in this plan and the Software Management Program Manual (SMPM).

The equivalent Software Change Management Plan shall be submitted to GEH for review and approval. GEH Supplier Design Services Document Review [2.3(2.c)] and GEH Supplier Supporting Document Review [2.3(2.m)] define the responsibilities and procedural requirements for review, approval, acceptance, and control of documentation or supporting documents from suppliers required for design services.

### **6.7.2 Acquired Software**

Acquired software is maintained and controlled in accordance with the procedures outlined in Subsection 3.9.2.

#### ***6.7.2.1 Acquired Software Configuration Change Control***

Acquired software may be modified by the supplier to:

- Correct discrepancies or deficient conditions
- Improve performance

If necessary, the RE shall reapply the evaluation process outlined in the SMPM [2.3(1)] to the modified acquired software.

After the required evaluation has been performed and the revised evaluation report and test results have been verified in accordance with the methods outlined in the SVVP, the acquired software, with its associated documentation package, shall be:

- Assigned a new revision number
- Baselined and placed under configuration control

## **6.8 RECORD COLLECTION AND RETENTION**

The baselined configuration items stored on a magnetic or optical medium shall undergo periodic archival backup in accordance with GEH Quality Record Computer Data [2.3(2.s)]. This document prescribes the requirements, procedures, and responsibilities for the control, retention, and retrieval of quality-related computer-based data maintained within the central computing facility of GEH. All configuration items shall contain a direct indication of the item's revision status.

## 7. SOFTWARE TEST PLAN

### 7.1 PURPOSE

The purpose of the Software Test Plan (STP) is to prescribe the scope, approach, resources, and schedule of the testing activities associated with the software development process. The Software Test Plan also identifies the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with this plan.

### 7.2 SCOPE

7.2.1 [[ ]]

[[

]]

7.2.2 [[ ]]

[[



]]

### **7.2.3 Test Submittal**

Test submittals include all test documentation and test items supplied by vendors. These submittals shall be transmitted, received, and approved in accordance with Subsection 3.9.1, Vendor Control.

7.3 [[ ]]

[[

]]

7.4 [[ ]]

[[

]]

7.5 [[

]]

[[

]]

7.6 [[

]]

[[

]]

7.7 [[

]]

[[

]]

7.7.1 [[        ]]

[[

]]

7.7.2 [[

]]

[[

]]

7.7.3 [[ ]]

[[

]]

## **8. SQAPM MAINTENANCE**

The SPE and SQA Manager are responsible for the maintenance of this Program. This SQAPM shall be assessed during the managerial review to ensure its suitability, adequacy, and effectiveness and revised to incorporate the agreed upon changes as described in Subsection 3.5.2. When improvements or deficiencies are identified, a Corrective Action Request (CAR) should be used to document the condition in accordance with GEH Self Assessment, Corrective Action and Audits [2.3(2.v)].

The CAR tracks activities and ensures that corrective and preventive actions are implemented. It ensures that the actions are effective in either eliminating the deficiency or improving the SQAPM. If a change to the SQAPM is warranted, one of the corrective activities shall determine if NRC notification is required and track the notification process as defined by the MMIS & HFE IP [2.1(1)]. The SQAPM shall be revised in accordance with the Design Change Control process described in Subsection 6.4.2.2. The SPE Manager or his designated delegate shall distribute the revised SQAPM to the organizations described in Subsection 3.2, Management Organization.

**9. TABLES & FIGURES**

**Table 1-1, Software Life Cycle Tasks, Responsibilities and Documentation-Planning Phase**

System Design Tasks	System Design Task Inputs	System Design Task Outputs	Development Organization	Review Organization	Associated Quality Tasks
[[					
					]]

Table 1-2, Requirements Phase

System Design Tasks	System Design Task Inputs	System Design Task Outputs	Development Organization	Review Organization	Associated Quality Tasks
[[					
					]]

Table 1-3, Design Phase

Software Design Tasks	Software Design Task Inputs	Software Design Task Outputs	Development Organization	Review Organization	Associated Quality Tasks
[[					

Software Design Tasks	Software Design Task Inputs	Software Design Task Outputs	Development Organization	Review Organization	Associated Quality Tasks

11

**Table 1-4, Implementation Phase**

Implementation Tasks	Implementation Task Inputs	Implementation Task Outputs	Implementation Organization	Review Organization	Associated Quality Tasks
[[					

Implementation Tasks	Implementation Task Inputs	Implementation Task Outputs	Implementation Organization	Review Organization	Associated Quality Tasks
					]]

**Table 1-5, Test Phase**

Test / Design Tasks	Test / Design Inputs	Test / Design Outputs	Design Organization	Test / Review Organization	Associated Quality Tasks
[[					
					]]

**Table 1-6, Installation Phase**

Test / Design Tasks	Test / Design Inputs	Test / Design Outputs	Design Organization	Test / Review Organization	Associated Quality Tasks
[[					

Test / Design Tasks	Test / Design Inputs	Test / Design Outputs	Design Organization	Test / Review Organization	Associated Quality Tasks



**Table 1-7, Operations and Maintenance Phase**

<b>Operation &amp; Maintenance Tasks</b>	<b>Operation &amp; Maintenance Task Inputs</b>	<b>Operation &amp; Maintenance Task Outputs</b>	<b>Operation &amp; Maintenance Organization</b>	<b>Review Organization</b>	<b>Associated Quality Tasks</b>
[[					
					]]

**Table 2, V&V and SSA Tasks Assigned to Each Software Class**

Table 2 lists the V&V and SSA tasks to be performed for software class Q, N3, and N2 software during the software life cycle phase. This table also indicates the organization responsible for conducting these tasks. If a V&V or SSA task is not pertinent to a particular software product, the V&V summary report shall contain the phrase, "The (task name) is not applicable to this (software product name)," with an appropriate reason for exclusion. The description of these V&V and SSA tasks are defined in Appendix E, V&V and SSA Tasks Description.

ESBWR Software Life Cycle Processes	Planning			Requirements			Design			Implementation			Test			Installation			Operations & Maintenance			
	Q	N3	N2	Q	N3	N2	Q	N3	N2	Q	N3	N2	Q	N3	N2	Q	N3	N2	Q	N3	N2	
Software Integrity Levels																						
[[																						











ESBWR Software Life Cycle Processes	Planning			Requirements			Design			Implementation			Test			Installation			Operations & Maintenance					
	Q	N3	N2	Q	N3	N2	Q	N3	N2	Q	N3	N2	Q	N3	N2	Q	N3	N2	Q	N3	N2			
Software Integrity Levels																								

**Table 3, Problems and Corrective Action Reporting**

Step	Scenario	Identified by	Documentation





**Table 4, Configuration Items**

Configuration Items	Format	Retention Medium
<b>Planning Phase</b>		
[[		
		]]
<b>Requirements Phase</b>		
[[		
		]]

Configuration Items	Format	Retention Medium
<b>Design Phase</b>		
[[		
		]]
<b>Implementation Phase</b>		
[[		
		]]
<b>Test Phase</b>		
[[		

Configuration Items	Format	Retention Medium
[[		
		]]
<b>Installation Phase</b>		
[[		
		]]
<b>Operation and Maintenance Phase</b>		
[[		
		]]

**Figure 4, Baseline Review Record**

This is an example of the form to be used for the Baseline Review Record.

**PLANNING BASELINE REVIEW RECORD**

**1st BASELINE**

**Revision 0**

PROJECT:		
PRODUCT:		DATE:

<b>CONFIGURATION MANAGEMENT:</b>	
OBJECTIVES:	
SCOPE:	
ITEMS TO BE BASELINED:	APPROVED DATE:
1.	
2.	
3.	

<b>V&amp;V AND SSA SUMMARY:</b>
ASSESSMENT:
RECOMMENDATION:

BASELINE REVIEW TEAM MEMBERS:
COMMENTS:
CONCLUSION:

Baseline Approved By Baseline Review Team Task Lead: \_\_\_\_\_  
*[Sign, Date and Print Name]*

**Figure 5, Software Library Structure**

Software library structure is dependant upon the medium and location of the library. Several software libraries for a single project may be required due to different media requirements or because of the use of COTS software or PDS. The following is an example of a structure of a Software Library located on a VAX development platform:

Directory Structure: [xxxxx.bbb.ccc]

where:

<b>Extension</b>	<b>Example</b>
xxxxx is the Product Type	PRM
bbb is the Category	BRR - Baseline Review Record SOURCECODE - Source Code etc.
ccc is the Released Software Revision	REV0 - Initial Software Release, REV1 - First Revision, etc.

For example:

PRM.SOURCECODE.REV0 is the directory location of Revision 0 of the Software Source code for the NUMAC Process Radiation Monitor (PRM).

For each software library used in the project, a supplemental document defining the software library structure shall be generated, stored in a DRF and linked to each appropriate DRF.

**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.

Conformance Code	Description
1	[[
2	
3	
4	]]

Appendix A - Software Plans Conformance Review								
Item	Reg Guide	IEEE Stand.	Related Software Plan			Deviation	Conform. Code	Justification
			SMPM	SQAPM	None			
NUREG 0800 BTP-14								
[[								

Regulatory Guides								

IEEE Standards from Section 2.2.4								




---

--	--	--	--	--	--	--	--	--

IEEE Standards from Section 2.4								

								]]

**Appendix B Acronyms and Abbreviations**

The following acronyms and abbreviations are used throughout this plan.

<b>Acronym</b>	<b>Meaning</b>
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
DRF	Design Record File
ECA	Engineering Change Authorization
ECN	Engineering Change Notice

Acronym	Meaning
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
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

Acronym	Meaning
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
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

Acronym	Meaning
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
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

<b>Acronym</b>	<b>Meaning</b>
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
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

**Appendix C Definitions**

<b>Term</b>	<b>Definition</b>
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].

Term	Definition
Build	An operational version of a system or component that incorporates a specified sub set of the capabilities that the final product will provide [IEEE 610.12].
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 or software and may be subdivided into other components [IEEE 610.12].
Computer language	A language designed to enable humans to communicate with computers [IEEE 610.12].
Configuration control	An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification [IEEE 610.12].
Configuration Item	An aggregation of hardware, software, design documents or procedures that is designated for configuration management and treated as a single entity in the configuration management process [IEEE 610.12].

Term	Definition
Criticality Analysis	<p>The degree of impact that a requirement, module, error, fault, failure, or other item has on the development or operation of a system.</p> <p>A method used to determine the impact of the software product on the system &amp; environment as a whole and thereby determine the software importance (i.e. safety-related, nonsafety-related, etc.).</p>
Design Documentation	<p>Design Documentation is information recorded about a specific life cycle activity. Documentation includes software life-cycle design outputs and software life cycle process documentation. A document may be in written or electronic format, and may contain text, illustrations, tables, computer files, program listings, binary images, and other forms of expression. A document for an activity may be packaged with documents for other activities, or documents for non-software life cycle activities. A document for an activity may be divided into several individual entities.</p>
Design output	<p>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&amp;M manuals, and training manuals.</p>
Design phase	<p>The <i>phase</i> in the software life cycle during which the designs for architecture, software components, interfaces, and data are created, documented, and verified to satisfy requirements [IEEE 610.12].</p>
Design Record File	<p>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.</p>
Design Reviews	<p>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.</p>
Deviation	<p>A departure from a specified requirement.</p>
Documentation	<p>A collection of documents on a given subject [IEEE 610.12].</p>

Term	Definition
Error	An incorrect step, process, or data definition [IEEE 610.12].
Failure Mode and Effects Analysis	A tabular method of providing traceability from the modes by which a system may fail and the effect of that failure on the ability of the system to perform its function, or the ability of a collection of systems to recover from the failure.
Fault Tree	A pictorial method of providing traceability from the modes by which a system may fail and the effect of that failure on the ability of the system to perform its function, or the ability of a collection of systems to recover from the failure.
Field Deviation Disposition Request	Field Deviation Disposition Request (FDDR) is used for documenting and disposition of the technical position for a deviation required in the field in supplied hardware, software, or services (see 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 <i>phase</i> in the software life cycle during which a software product is created from design documentation and debugged [IEEE 610.12].
Independent Verification and Validation (IV&V)	Verification and Validation performed by an Organization that is technically managerially and financially independent of the Organization [IEEE 610.12] and RG 1.168 Section C3 [2.2.3(1)].
Installation Phase	The <i>phase</i> in the software life cycle during which the software product is installed into its operational environment and tested to ensure that it performs as intended [IEEE 610.12].
Instrument	A hardware device used for analytical or control functions and usually containing an embedded microprocessor(s).

Term	Definition
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	<p>1) A shared boundary across which information is passed. This definition is interpreted broadly to include design interfaces between participating design organizations.</p> <p>2) A hardware or software component that connects two or more other components for the purpose of passing information from one to the other.</p> <p>3) To connect two or more components for the purpose of passing information from one to the other.</p> <p>4) To serve as a connecting or connected component as in (2). [IEEE 610.12 as modified by RG 1.69</p>
Metric	A quantitative measure of the degree to which a system, component, or process possesses a given attribute [IEEE 610.12].
Module	A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to, or output from, and assembler, compiler, linkage editor, or executive routine [IEEE 610.12].
Operations and Maintenance Phase	The <i>phase</i> in the software life cycle during which the software product is functioning in its operational environment, monitored for satisfactory performance and modified as necessary to correct problems or to respond to changing requirements [IEEE 610.12].
Package	A separately compilable software component consisting of related data types, data objects and sub-programs [IEEE 610.12].
Path Testing	Testing designed to execute all or selected paths through a computer program [IEEE 610.12].
Planning Phase	The initial <i>phase</i> of a software development project, in which project scope, purpose, strategy, schedule and milestones are established and user needs through documentation (for example, system definition documentation and procedures) are described and evaluated.
Procedure	A course of action to be taken to perform a given task [IEEE 610.12].

Term	Definition
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	<p>A condition or capability that must be met or possessed by a system or system component to satisfy a contract standard specification or other formally imposed documents [IEEE 610.12].</p> <p>In specifying requirements, the word <b>shall</b> is used to indicate mandatory requirements and from which no deviation is permitted (<b>'shall'</b> and <b>'required to'</b> are equivalent in meaning).</p> <p>Requirements are not specified with the word <b>should</b>. 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 necessarily required; Also, that (in the negative form) a certain course of action is not prohibited (<b>'should'</b> and <b>'recommended'</b> are equivalent in meaning).</p>
Requirements Phase	The <i>phase</i> in the software life cycle during which the requirements for a software product are defined and documented [IEEE 610.12].
Requirements 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.

Term	Definition
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].
Safety-Related	<p>Safety-related structures, systems, components, and parts provide safety-related functions necessary to assure:</p> <ul style="list-style-type: none"> <li>a. The integrity of the reactor coolant pressure boundary; or</li> <li>b. The capability to shut down the reactor and maintain it in a safe shutdown condition; or</li> <li>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.</li> </ul>
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	<p>Nonsafety-related systems software whose failure could challenge safety systems as defined below:</p> <ul style="list-style-type: none"> <li>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)].</li> <li>b. Software that is intended to mitigate the result of an accident.</li> <li>c. Software that is intended to recover from the result of an accident.</li> </ul>
Software Class Q	Software performs functions classified per GEH Safety-Related Classification determination process [2.3(2.t)] as safety-related.

Term	Definition
Software Development Process	The process by which user needs are translated into a software product. The process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code, and sometimes, installing and checking out the software for operational use [IEEE 610.12].
Software Feature	A distinguishing characteristic of a software item, such as, performance, portability, or functionality.
Software Item	Source code, object code, job control code, control data, or a collection of these items [IEEE 610.12].
Software Life cycle	The period of time that begins when a software product is conceived and ends when the software is no longer available for use [IEEE 610.12].
Software Life cycle Phase	The division of the software life cycle into discrete logical units. The I&C software life cycle is divided into eight <i>phases</i> , 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 of 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].

<b>Term</b>	<b>Definition</b>
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 developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement [IEEE 610.12].
Test Item	A software item that is an object of testing [IEEE 610.12].
Test Log	A chronological record of all relevant details about the execution of a test [IEEE 610.12].
Test Objective	An identified set of software features to be measured under specified conditions by comparing actual behavior with the required behavior described in the software documentation [IEEE 610.12].
Test Phase	The <i>phase</i> in the software life cycle during which the components of a software product are integrated with the hardware and evaluated to determine whether or not performance requirements have been satisfied [IEEE 610.12].
Test Plan	A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do such task, and any risks requiring contingency planning [IEEE 610.12].
Traceability Matrix	A matrix that records the relationship between two or more product specifications (i.e., design documentation) of the development process (e.g., a matrix that records the relationship between the requirements and the design of a given software component) [IEEE 610.12].
Unit 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].

<b>Term</b>	<b>Definition</b>
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.

**Appendix D Software Characteristics**

Software characteristics important to safety system software as defined by NUREG 0800, SRP, [2.2.1(1)]. These characteristics are divided into two sets:

- Software functional characteristics
- Software development process characteristics

<b>Functional Term</b>	<b>Definition</b>
Accuracy	The degree of freedom from error of sensor and operator input, the degree of exactness exhibited by an approximation or measurement, and the degree of freedom from error of actuator output.
Functionality	The operations, which must be carried out by the software. Functions generally transform input information into output information in order to affect the reactor operation. Inputs may be obtained from sensors, operators, other equipment, or other software. Outputs may be directed to actuators, operators, other equipment, or other software.
Reliability	The degree to which a software system or component operates without failure. This definition does not consider the consequences of failure, only the existence of failure.
Robustness	The ability of a software system or component to function correctly in the presence of invalid inputs or stressful environmental conditions. This includes the ability to function correctly despite some violation of the assumptions in its specification.
Safety	Those properties and characteristics of the software system that directly affect or interact with system safety considerations. The other characteristics discussed in Chapter 7 of NUREG 0800 SRP [2.2.1(1)] are important contributors to the overall safety of the software-controlled safety system, but are primarily concerned with the internal operation of the software. The safety characteristic, however, is primarily concerned with the effect of the software on system hazards and the measures taken to control those hazards.
Security	The ability to prevent unauthorized, undesired, and unsafe intrusions. Security is a safety concern insofar as such intrusions can affect the safety-related functions of the software.
Timing	The ability of the software system to achieve its timing objectives within the hardware constraints imposed by the computing system being used.

<b>Functional Term</b>	<b>Definition</b>
Completeness	Those attributes of the planning documents, implementation process documents and design outputs that provide full implementation of the functions required of the software. The functions, which the software is required to perform are derived from the general functional requirements of the safety system, and the assignment of functional requirements to the software in the overall system design.
Consistency	The degree of freedom from contradiction among the different documents and components of a software system. There are two aspects to consistency. Internal consistency denotes the consistency within the different parts of a component for example; a software design is internally consistent if no set of design elements are mutually contradictory. External consistency denotes the consistency between one component and another for example, software requirements and the resulting code are consistent with one another if there are no contradictions between the requirements and the code.
Correctness	The degree to which a design output is free from faults in its specification, design, and implementation. There is considerable overlap between correctness properties and properties of other characteristics such as accuracy and completeness.
Style	The form and structure of a planning document, implementation process document or design output. Document style refers to the structure and form of a document. This has connotations of understandability, readability, and modifiability. Programming style refers to the programming language characteristics of the software and programming techniques, which are mandated, encouraged, discouraged, or prohibited in a given implementation.
Traceability	The degree to which each element of one life cycle product can be traced forward to one or more elements of a successor life cycle product, and can be traced backwards to one or more elements of a predecessor life cycle product.
Unambiguity	The degree to which each element of a product, and of all elements taken together, have only one interpretation.
Verifiability	The degree to which a software planning document, implementation process document or design output is stated or provided in such a way as to facilitate the establishment of verification criteria and the performance of analyses, reviews, or tests to determine whether those criteria have been met.

**Appendix E V&V Tasks Definitions**

<b>Term</b>	<b>Definition</b>
Algorithm Analysis	Verify the correct implementation of algorithms. Equations, mathematical formulations, or expressions. Rederive any significant algorithms, and equations from basic principles and theories. Compare against established references or proven past historical data. Validate the algorithms, equations, mathematical formulations, or expressions with respect to the system and software requirements. Ensure that the algorithms and equations are appropriate for the problem solution. Validate the correctness of any constraints or limitations such as rounding, truncation, expression simplifications, best-fit estimations, non-linear solutions imposed by the algorithms and equations [IEEE-1012].
Anomaly Evaluation	Assessment of software that deviates from documented requirements, specifications, design, user documents, or standards. The assessment should include risk based on probability and severity of occurrence. [IEEE-1012].
Audit Performance	Provide an independent assessment of whether a software process and its products conform to applicable regulations, standards, plans, procedures, specifications and guidelines. Audits may be applied to any software process or product at any development stage. Audits may be initiated by the supplier, the acquirer, the developer or other involved party such as a regulatory agency. The initiator of the audit selects the audit team and determines the degree of independence required. The initiator of the audit and the audit team leader establish the purpose, scope, plan, and reporting requirements for the audit. The auditors collect sufficient evidence to decide whether the software processes and products meet the evaluation criteria. They identify major deviations, assess risk to quality, schedule, and cost and then report their findings. Examples of processes that could be audited include configuration management practices, use of software tools, degree of integration of the various software engineering disciplines particularly in developing architecture, security issues, training, project management [IEEE-1012].
Concept Documentation Evaluation	Verify that the concept documentation satisfies user needs and is consistent with acquisition needs. Validate constraints of interfacing systems and constraints or limitations of proposed approach [IEEE-1012].

Term	Definition
Control Flow Analysis	Assess the correctness of the software by diagramming the logical control. Examine the flow of the logic to identify missing, incomplete, or inaccurate requirements. Validate whether the flow of control amongst the functions represents a correct solution to the problem [IEEE-1012].
Criticality Analysis	A structured evaluation of the software characteristics (e.g., safety, security, complexity, performance) for severity of impact of system failure, system degradation, or failure to meet software requirements or system objectives [IEEE-1012].
Database Analysis	<p>Evaluate database design as part of a design review process could include the following:</p> <p><b>Physical Limitations Analysis</b> Identify the physical limitations of the database such as maximum number of records, maximum record length, largest numeric value, smallest numeric value, and maximum array length in a data structure and compare them to designed values.</p> <p><b>Index vs. Storage Analysis</b> Analyze the use of multiple indexes compared to the volume of stored data to determine if the proposed approach meets the requirements for data retrieval performance and size constraints.</p> <p><b>Data Structures Analysis</b> Some database management systems have specific data structures within a record such as arrays, tables, and date formats. Review the use of these structures for potential impact on requirements for data storage and retrieval.</p> <p><b>Backup and Disaster Recovery Analysis</b> Review the methods employed for backup against the requirements for data recovery and system disaster recovery and identify deficiencies [IEEE-1012].</p>

Term	Definition
Data Flow Analysis	<p>Evaluate data flow diagrams as part of a design review process. This could include the following:</p> <p><b>Symbology Consistency Check.</b> The various methods used to depict data flow diagrams employ very specific symbology to represent the actions performed. Verify that each symbol is used consistently.</p> <p><b>Flow Balancing.</b> Compare the output data from each process block to the data inputs and the data derived within the process to ensure that data is available when required. This process does not specifically examine timing of sequence considerations.</p> <p><b>Confirmation of Derived Data.</b> Examine the data derived within a process for correctness and format. Data designed to be entered into a process by operator action should be confirmed to ensure availability.</p> <p><b>Keys to Index Comparison.</b> Compare the data keys used to retrieve data from data stores within a process to the database index design to confirm that no invalid keys have been used and the uniqueness properties are consistent [IEEE-1012].</p>
Design Phase Inspection	Design Phase Baseline Review
Disaster Recovery Plan Assessment	<p>Verify that the disaster recovery plan is adequate to restore critical operation of the system in the case of an extended system outage. The disaster recovery plan should include the following:</p> <ul style="list-style-type: none"> <li>• Identification of the disaster recovery team and a contact list.</li> <li>• Recovery operation procedures.</li> <li>• Procedure for establishing an alternative site including voice and data communications, mail, and support equipment.</li> <li>• Plans for replacement of computer equipment</li> <li>• Establishment of a system backup schedule</li> <li>• Procedures for storage and retrieval of software, data, documentation, and vital records off-site.</li> <li>• Logistics of moving staff, data, documentation, etc [IEEE-1012].</li> </ul>

<b>Term</b>	<b>Definition</b>
Distributed Architecture Assessment	Assess the distribution of data and processes in the proposed architecture for feasibility, timing compliance, availability of telecommunications, cost, backup and restore features, downtime, system degradation, and provisions for installation of software updates [IEEE-1012].
Evaluation of New Constraints	Evaluate new constraints (e.g., operational requirements, platform characteristics, operating environment) on the system or software requirements to verify the applicability of the SVVP. Software changes are maintenance activities [IEEE-1012].
Hazard Analysis	A systematic qualitative or quantitative evaluation of software for undesirable outcomes resulting from the development or operation of a system. These outcomes may include injury, illness, death, mission failure, economic loss, property loss, environmental loss, or adverse social impact. This evaluation may include screening or analysis methods to categorize, eliminate, reduce, or mitigate hazards [IEEE-1012].
Independent Risk Assessment	Conduct an independent risk assessment on any aspect of the software project and report on the findings. Such risk assessments will be primarily from a system perspective. Examples of risk assessment include appropriateness of the selected development methodology or tools for the project; and quality risks associated with proposed development schedule alternatives [IEEE-1012].
Installation Checkout Report Evaluation	Conduct analyses or test to verify that the installed software corresponds to the software subjected to V&V. Verify that the software code and databases initialize, execute, and terminate as specified. In the transition from one version of software to the next, the V&V efforts shall validate that the software can be removed from system to without affecting the functionality of the remaining system components. The V&V effort shall verify the requirements for continuous operation and service during transition, including user notification.
Installation Configuration Audit	Verify that all software products required to correctly install and operate the software are present in the installation package. Validated that all site dependent parameters or conditions to verify supplied values are correct.
Interface Analysis	Verify and validate that the requirements for software interfaces with hardware, user operator and other systems are correct, consistent, complete, accurate, and testable [IEEE-1012].

<b>Term</b>	<b>Definition</b>
Implementation Phase Inspection	Implementation Phase Baseline Review
Operation Procedures Evaluation	Verify that the operating procedures are consistent with the user documentation and conform to the system requirements.
Planning Phase Inspection	Planning Phase Baseline Review
Planning the Interface between the V&V Effort and Supplier	Plan the V&V schedule for each V&V task. Identify the preliminary list of development processes and products to be evaluated by the V&V processes. Describe V&V access rights to proprietary and classified information. It is recommended that the plan be coordinated with the acquirer. Incorporate the project software integrity level scheme into the planning process [IEEE-1012].
Previously Developed Software Assessment	<p>Only formal assessments of existing software will be addressed.</p> <p>Assessment of existing software is an iterative assessment (the comparison of the software in the same domain over time or comparative to other domains within the same existing software being studied).</p> <p>The assessment can be formative, summative, objective, subjective, criterion-referenced, and/or norm-referenced.</p>
Project Management Oversight Support	Assess project development status for technical and management issues, risks, and problems. Coordinate oversight assessment with the acquirer and development organization. Evaluate project plans, schedules, development processes, and status. Collect, analyze, and report on key project metrics [IEEE-1012].
Requirements Phase Inspection	Requirements Phase Baseline Review
Risk Analysis	The systematic use of available information to identify hazards and estimate the risk to individuals or populations, property or the environment [IEEE-1012 Annex I].

Term	Definition
Security Assessment	<p>Evaluate the security controls on the system to ensure that they protect the hardware and software components from unauthorized use, modifications, and disclosures, and to verify the accountability of the authorized users. Verify that these controls are appropriate for achieving the system's security objectives. A system security assessment should include both the physical components (e.g., computers, controllers, networks, modems, radio frequency, infrared devices) and logical components (e.g., operating systems, utilities, application programs, communication protocols, data, administrative operating policies and procedures). [IEEE-1012]</p>
Simulation Analysis	<p>Use a simulation to exercise the software or portions of the software to measure the performance of the software against predefined conditions and events. The simulation can take the form of a manual walkthrough of the software against specific program values and inputs. The simulation can also be another software program that provides the inputs and simulation of the environment to the software under examination. Simulation analysis is used to examine critical performance and response time requirements or the software's response to abnormal events and conditions [IEEE-1012].</p>
Sizing and Timing Analysis	<p>Collect and analyze data about the software functions and resource utilization to determine if system and software requirements for speed and capacity are satisfied. The types of software functions and resource utilization issues include, but are not limited to the following:</p> <ul style="list-style-type: none"> <li>• CPU load.</li> <li>• Random access memory and secondary storage (e.g., disk, tape) utilization.</li> <li>• Network speed and capacity.</li> <li>• Input and output speed.</li> <li>• Sizing and timing analysis is started at software design and iterated through acceptance testing [IEEE-1012].</li> </ul>
Software Design Evaluation	<p>Evaluate the design elements (SDD) for correctness, consistency, completeness, accuracy, readability, and testability [IEEE-1012].</p>

Term	Definition
Software Regression Analysis and Testing	Determine the extent of V&V analysis and tests that must be repeated when changes are repeated when changes are made to any previously examined software products. Assess the nature of the change to determine ripple or side effects and impacts on other aspects of the system. Rerun test cases based on changes, error corrections, and impact assessment, to determine errors spawned by software modifications. [IEEE 1012 Annex G]
Software Requirements Evaluation	Evaluation of the essential requirements (i.e., functions, performance, design constraints, and attributes) of the software.
Software V&V Plan (SVVP) Generation	Generate and SVVP for all life cycle processes. The SVVP may require updating throughout the life cycle. Outputs of other activities are inputs to the SVVP. Establish a baseline SVVP prior to the requirements V&V activities. Identify project milestones in the SVVP. Schedule V&V tasks to support project managements reviews and technical reviews [IEEE-1012].
Source Code and Source Code Documentation Evaluation	Evaluate the source code components (Source Code Documentation) for correctness, consistency, completeness, accuracy, readability, and testability [IEEE-1012].
Support Tool Evaluation	The systematic determination of merit, worth, and significance of a programming tool. Support tool is a program or application used to create, debug, or maintain other programs and applications.
System Software Assessment	Assess system software (e.g., operating system, computer-aided software engineering tools, data base management system, repository, telecommunications software, graphical user interface) for feasibility, impact on performance and functional requirements, maturity, supportability, adherence to standards, developer's knowledge of and experience with the system software and hardware, and software interface requirements [IEEE-1012].
Test Certification	Certify the test results by verifying that the tests were conducted using baselined requirements, a configuration control process, and repeatable tests, and by witnessing the tests. Certification may be accomplished at a software configuration item level or at a system level [IEEE-1012].
Test Phase Inspection	Test Phase Baseline Review

<b>Term</b>	<b>Definition</b>
Test Witnessing	Monitor the fidelity of test execution to the specified test procedures, and witness the recording of test results. When a test failure occurs, the testing process can be continued by 1) implementing a “workaround” to the failure; 2) inserting a temporary code patch; or 3) halting the testing process and implementing a software repair. In all cases, assess the test continuation process for test process breakage (e.g., some software is not tested or a patch is left in place permanently), adverse impact on other, tests and loss of configuration control. Regression testing should be done for all the software affected by the test failure [IEEE-1012].
Traceability Analysis	Trace the software requirements (SRS and HHS) to system requirements (SDS) and system requirements to software requirements. Analyze identified relationships to correctness, consistency, completeness, and accuracy [IEEE-1012].

**MFN 08-256**

**Enclosure 3**

**Affidavit**

# GE-Hitachi Nuclear Energy LLC

## AFFIDAVIT

I, **David H. Hinds**, state as follows:

- (1) I am the General Manager, New Units Engineering, GE-Hitachi Nuclear Energy LLC (GEH) have been delegated the function of reviewing the information described in paragraph (2) which is sought to be withheld, and have been authorized to apply for its withholding.
- (2) The information sought to be withheld is contained in Enclosure 1 of GEH letter MFN 08-256, Mr. Richard E. Kingston to U.S. Nuclear Regulatory Commission, entitled *Submittal of ESBWR Licensing Topical Report – ESBWR Software Quality Assurance Plan Manual*, Revision 3 (NEDE-33245P) – GEH Proprietary Information, dated July 11, 2008 is delineated by a [[dashed underline inside double square brackets.<sup>{3}</sup>]]. Figures and large equation objects are identified with double square brackets before and after the object. In each case, the superscript notation <sup>{3}</sup> refers to Paragraph (3) of this affidavit, which provides the basis for the proprietary determination.
- (3) In making this application for withholding of proprietary information, of which it is the owner, GEH relies upon the exemption from disclosure set forth in the Freedom of Information Act ("FOIA"), 5 USC Sec. 552(b)(4), and the Trade Secrets Act, 18 USC Sec. 1905, and NRC regulations 10 CFR 9.17(a)(4), and 2.390(a)(4) for "trade secrets" (Exemption 4). The material for which exemption from disclosure is here sought also qualifies under the narrower definition of "trade secret", within the meanings assigned to those terms for purposes of FOIA Exemption 4 in, respectively, Critical Mass Energy Project v. Nuclear Regulatory Commission, 975F2d871 (DC Cir. 1992), and Public Citizen Health Research Group v. FDA, 704F2d1280 (DC Cir. 1983).
- (4) Some examples of categories of information which fit into the definition of proprietary information are:
  - a. Information that discloses a process, method, or apparatus, including supporting data and analyses, where prevention of its use by GEH competitors without license from GEH constitutes a competitive economic advantage over other companies;
  - b. Information which, if used by a competitor, would reduce his expenditure of resources or improve his competitive position in the design, manufacture, shipment, installation, assurance of quality, or licensing of a similar product;
  - c. Information which reveals aspects of past, present, or future GEH customer-funded development plans and programs, resulting in potential products to GEH;
  - d. Information which discloses patentable subject matter for which it may be desirable to obtain patent protection.

The information sought to be withheld is considered to be proprietary for the reasons set forth in paragraphs (4)a., and (4)b., above.

- (5) To address 10 CFR 2.390 (b) (4), the information sought to be withheld is being submitted to NRC in confidence. The information is of a sort customarily held in confidence by GEH, and is in fact so held. The information sought to be withheld has, to the best of my knowledge and belief, consistently been held in confidence by GEH, no public disclosure has been made, and it is not available in public sources. All disclosures to third parties including any required transmittals to NRC, have been made, or must be made, pursuant to regulatory provisions or proprietary agreements, which provide for maintenance of the information in confidence. Its initial designation as proprietary information, and the subsequent steps taken to prevent its unauthorized disclosure, are as set forth in paragraphs (6) and (7) following.
- (6) Initial approval of proprietary treatment of a document is made by the manager of the originating component, the person most likely to be acquainted with the value and sensitivity of the information in relation to industry knowledge. Access to such documents within GEH is limited on a "need to know" basis.
- (7) The procedure for approval of external release of such a document typically requires review by the staff manager, project manager, principal scientist or other equivalent authority, by the manager of the cognizant marketing function (or his delegate), and by the Legal Operation, for technical content, competitive effect, and determination of the accuracy of the proprietary designation. Disclosures outside GEH are limited to regulatory bodies, customers, and potential customers, and their agents, suppliers, and licensees, and others with a legitimate need for the information, and then only in accordance with appropriate regulatory provisions or proprietary agreements.
- (8) The information identified in paragraph (2), above, is classified as proprietary because it identifies details of GEH ESBWR methods, techniques, information, procedures, and assumptions related to the application of the software plans to the GEH ESBWR.

The development of the evaluation process along with the interpretation and application of the regulatory guidance is derived from the extensive experience database that constitutes a major GEH asset.

- (9) Public disclosure of the information sought to be withheld is likely to cause substantial harm to GEH's competitive position and foreclose or reduce the availability of profit-making opportunities. The information is part of GEH's comprehensive BWR safety and technology base, and its commercial value extends beyond the original development cost. The value of the technology base goes beyond the extensive physical database and analytical methodology and includes development of the expertise to determine and apply the appropriate evaluation process. In addition, the technology base includes the value derived from providing analyses done with NRC-approved methods.

The research, development, engineering, analytical and NRC review costs comprise a substantial investment of time and money by GEH.

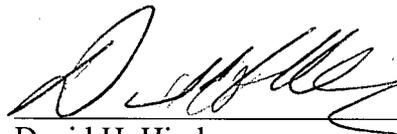
The precise value of the expertise to devise an evaluation process and apply the correct analytical methodology is difficult to quantify, but it clearly is substantial.

GEH's competitive advantage will be lost if its competitors are able to use the results of the GEH experience to normalize or verify their own process or if they are able to claim an equivalent understanding by demonstrating that they can arrive at the same or similar conclusions.

The value of this information to GEH would be lost if the information were disclosed to the public. Making such information available to competitors without their having been required to undertake a similar expenditure of resources would unfairly provide competitors with a windfall, and deprive GEH of the opportunity to exercise its competitive advantage to seek an adequate return on its large investment in developing these very valuable analytical tools.

I declare under penalty of perjury that the foregoing affidavit and the matters stated therein are true and correct to the best of my knowledge, information, and belief.

Executed on this 11<sup>th</sup> day of July 2008.

A handwritten signature in black ink, appearing to read "D. Hinds", written over a horizontal line.

David H. Hinds  
GE-Hitachi Nuclear Energy LLC