Enclosure 4

MFN 10-043

Submittal of Markups to Licensing Topical Reports Related to ESBWR Design Certification -

NEDO-33245, Software Quality Assurance Program Manual, Revision 5

Change List and Markups

Public Version

HITACHI

GE Hitachi Nuclear Energy

NEDO-33245
Revision 5
Class I
DRF 0000-0049-7141
February 2010

Licensing Topical Report

ESBWR – SOFTWARE QUALITY ASSURANCE PROGRAM MANUAL

NON-PROPRIETARY INFORMATION NOTICE

This is a non-proprietary version of NEDE-33245, Rev 5, which has the proprietary information removed. Portions of the document that have been removed are indicated by open and closed double square 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.

Table of Contents

List of Tables

List of Figures

SUMMARY OF CHANGES FROM PREVIOUS REVISION

ITEM	LOCATION	CHANGE
Entire Document		
1.	Entire document	Minor editorial changes for legibility.
Section 1		
2.	Section 1.1, 1st para.	Deleted "I&C system" and replaced with "digital computer-based plant process control and monitoring software" to incorporate changes per RAI 7.1-142.
3.	Section 1.2, Last para.	Added two phrases; "digital computer-based plant process control and monitoring software. This includes" and "It also includes non-DCIS real time plant systems such as but not limited to local fire protection systems, local programmable logic controllers (PLCs), digital standalone controllers and indicators, inverters and battery chargers, electrical distribution digital protective relays, switchgear instrumentation and circuit breaker controllers, meteorological monitoring systems, digital hygrometers, digital salinity cells, digital pH meters, digital conductivity cells, digital dissolved gas monitors, digital area explosive gas monitors, etc." to incorporate changes per RAI 7.1-142.
4.	Section 1.5, 2nd to last para.	Deleted "I&C system" and replaced with "digital computer-based plant process control and monitoring software" to incorporate changes per RAI 7.1-142.
5.	Section 1.5, Fig. 1	Corrected reference in 1st diamond of figure 1 and added cross-reference to Note 1 for consistency's sake.
Section 2		
6.	Section 2.3, 2nd List, GE Hitachi Nuclear Energy Procedures and Policies, Reference Number, 2.u	Changed reference from "Dedication of Commercial Grade Items" to "Commercial Grade Dedication of Software and Digital Components with Embedded Software for Use in Safety-Related Instrumentation and Control Applications, in Accordance with EPRI-TR-106439-1996" to incorporate changes per RAI 7.1-142.

ITEM	LOCATION	CHANGE
7.	Section 2.3, 2nd List, GE Hitachi Nuclear Energy Procedures and Policies, Reference Number, 2.z	Added new reference "Qualification of Previously Developed Software" to incorporate changes per RAI 7.1-142.
Section 3		
8.	Section 3.1, 1st para.	Deleted "I&C system" and replaced with "digital computer-based plant process control and monitoring software" to incorporate changes per RAI 7.1-142.
9.	Sub-section 3.2.3, 1st para., 1st sentence	Changed "experience" to "experienced" for grammatical correctness.
Section 4		
10.	Sub-section 4.2.4	Corrected cross-references within sub-bullets 11 through and including 15.
11.	Sub-section 4.3.3.1, Table 4.3.3.1-1	Made a grammatical correction to the SSA Output for Task 5: changed the word "doest" to the words "does not".
12.	Sub-section 4.3.3.1, Table 4.3.3.1-1	Made a grammatical correction to the SSA Output for Task 8: changed the word "riming" to the word "timing".
13.	Section 4.4, 1st sentence	Changed the article "The" to the demonstrative pronoun "This" for the sake of grammatical correctness.
14.	Sections 4.5 through 4.10	For formatting consistency, renumbered sections 4.5 through 4.9 to be underneath section 4.4. Renumbered section 4.10 to section 4.5.
Section 5		
15.	Sub-section 5.3.1.1, 1st sentence, 1st para.	Deleted the word "it" for grammatical correctness.
16.	Sub-section 5.3.1.6, Table 5.3.1.6-1	In the activity for task 3, specified that system requirements review is performed of System Design Specification for consistency with Table 5.6-1 in SMPM [2.3(1.a)].

ITEM	LOCATION	CHANGE
17.	Sub-section 5.3.1.6, Table 5.3.1.6-1	Added a note to specify who performs V&V activities for N3 and N2 acquired software consistent with section 3.9.
18.	Sub-section 5.3.2.1, Table 5.3.2.1-1	In last bullet of the outputs, changes sentence for grammatical correctness.
19.	Sub-section 5.3.4.3, 1st para.	Deleted reference to component testing when SVT and SFAT are being performed and changed to system testing for consistency with sections 7.3 and 7.4. Component level testing is performed to the SFT in accordance with SMPM [2.3(1.a)], sub-section 6.11.2.1.
20.	Sub-section 5.3.5.4, 1st sentence	Corrected criticality cross-reference from sub-section 4.3.3.1 to 4.3.4.1.
21.	Sub-section 5.3.7.6, 1st sentence Table 5.3.7.6-1, Task 1	Added phraseology to indicate task includes both the generation and the evaluation of the Installation Configuration Table for the sake of consistency with the title of this sub-section. Originally the verbiage indicated just evaluation, NOT generation.
22.	Sub-section 5.3.7.8, 1st sentence Table 5.3.7.8-1, 1st Task	Revised verbiage to include both hazard and risk analysis consistent with title of sub-section.
23.	Sub-section 5.3.8.8, 1st sentence 2nd and 3rd sentences, and Table 5.3.8.8-1	Corrected cross-reference in 1st sentence from sub-section 4.3.6 to sub-section 4.3.7. Revised verbiage to include both hazard and risk analysis consistent with title of sub-section.
24.	Sub-section 5.3.8.9, 1st sentence	Changed the verb "identify" to the past participle "identified" for grammatical correctness.
25.	Sub-section 5.4.3	Corrected cross-reference to test report from sub-section 7.7.3 to sub-section 7.7.8.

ITEM	LOCATION	CHANGE
26.	Sub-section 5.3.9.2. 1st para.	Changed text from "Dedication of Commercial Grade Items" to "Commercial Grade Dedication of Software and Digital Components with Embedded Software for Use in Safety-Related Instrumentation and Control Applications, in accordance with EPRI-TR-106439-1996" and provided appropriate Sub-section 2.3 Table reference (2.u) to incorporate changes per RAI 7.1-142.
27.	Sub-section 5.3.9.2. 2nd para.	Added new paragraph: "Applicable portions of Commercial Grade Dedication of Software and Digital Components with Embedded Software for Use in Safety-Related Instrumentation and Control Applications, in accordance with EPRI-TR-106439-1996 may be used for guidance in evaluating commercial off the shelf software designated for use in nonsafety-related applications." Note that the phrase "testing and qualifying" used in the response to RAI 7.1-142 is changed to the more correct "evaluating" as identified during review of corresponding sub-section 5.8.3.6 of revision 5 to the SMPM [2.3(1.a)].
28.	Sub-section 5.3.9.3. 1st para.	Added text "Qualification of Previously Developed Software" and provided appropriate Sub-section 2.3 Table reference (2.z) to incorporate changes per RAI 7.1-142.
29.	Sub-section 5.3.9.3. 2nd para.	Added new paragraph: "Applicable portions of Qualification of Previously Developed Software [2.3(2.z)] may be used to evaluate the PDS for applicability in nonsafety-related applications."
Section 6		
30.	Sub-section 6.4.1, 4th bullet, 2nd para.	Changed "files" to "file" for grammatical correctness.
31.	Subsection 6.4.2.2, Table 6.4.2.2-2	Grammatically corrected the "Change Process Steps" cell in the responsible individual row (2nd row from last).
Section 7		

ITEM	LOCATION	CHANGE
32.	None	None
Section 8		
33.	Section 8, 1st para., 2nd sentence	Corrected cross-reference from subsection 3.5.2 to 3.5.1.2.
Section 9		
34.	Tables 9-1a through g, last row	The system design task associated with the quality task "Planning Baseline Review", "Requirements Baseline Review", etc., was not specified; this is now corrected in Tables 9-1a through g be "Baseline Review".
35.	Table 9-1b, 5th row	Review organization cell for the system design task of "Develop SyRS RTA" was NOT specified for various classes of software; this is now corrected.
36.	Table 9-1c, 6th row	Development organization for the SVT development task was specified to be SPE instead of the IVVT within SPE; this is now corrected.
Section 10		
37.	None	None
Appendices		
38.	Appendix C, Terms	Added definition for "Plant Process Control and Monitoring Software" to incorporate changes per RAI 7.1-142.

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 Nuclear Safety-Related (Quality Class Q) and Nonsafety-Related (Quality Class N3 and N2) digital computer-based plant process control and monitoring software, 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] and Branch Technical Position (BTP) HICB-14 R4, Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems [2.2.1], 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 a 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 [2.2.4].
- 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 Policies and Procedures (P&Ps) to guide software development
  - The Engineering Operating Procedures (EOPs)
  - The requirements described in this SQAPM
  - The Software Management Program Manual (SMPM) [2.3(1.a)] (hereafter referred to as SMPM)
- The design documentation and design outputs for each software life cycle phase defined in the SMPM [2.3(1.a)] 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:

- 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 Test Program [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 digital computer-based plant process control and monitoring software. This includes the I&C systems, as specifically defined in the MMIS/HFE IP [2.1] (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. It also includes non-DCIS real time plant systems such as but not limited to local fire protection systems, local programmable logic controllers (PLCs), digital standalone controllers and indicators, inverters and battery chargers, electrical distribution digital protective relays, switchgear instrumentation and circuit breaker controllers, meteorological monitoring systems, digital hygrometers, digital salinity cells, digital pH meters, digital conductivity cells, digital dissolved gas monitors, digital area explosive gas monitors, etc.

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 GEH vendors shall comply with this SQAPM. If a vendor elects to follow its established SQA program, then the SQA program as defined in the purchase order (Section 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

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

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

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 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. 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 safety-related functions as specified by the Safety-Related Classification determination process [2.3(2.t)].
Software Class N3	Nonsafety-related systems software whose failure could challenge safety-related systems as defined below: • 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] • Software that is intended to mitigate the result of an accident • Software that is intended to support recovery from the result of an accident
Software Class N2	• Software failure cannot adversely affect a safety-related function • Software failure results in inconvenience to the user

[[ ]]

Figure 1. Software Class Evaluation Process

[[

]]

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 1.152-2006, Criteria for Use of Computers in Safety Systems of Nuclear Power Plants
- RG 1.168-2004, Verification, Validation, Reviews, and Audits For Digital Computer Software Used in Safety Systems of Nuclear Power Plants
- RG 1.169-1997, Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants
- RG 1.170-1997, Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants
- RG 1.171-1997, Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants
- RG 1.172-1997, Software Requirements Specifications for Digital Computer Software Used in Safety Systems of Nuclear Power Plants
- RG 1.173-1997, Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants

2.2.4 Institute of Electrical and 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 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.a)] and this SQAPM, and are potentially subject to NRC notification. The NRC notification process is addressed in the MMIS/HFE Implementation Plan [2.1].

- IEEE 7-4.3.2-2003, IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations
- IEEE 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	Applicable LTRs	Document Number
1.a	ESBWR Software Management Program Manual (SMPM)	NEDO-33226
1.b	ESBWR Cyber Security Program Plan	NEDO-33295

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

GE Hitachi Nuclear Energy Procedures and Policies		
Reference Number	**Document Title**	**Abstract**
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.

GE Hitachi Nuclear Energy Procedures and Policies		
Reference Number	**Document Title**	**Abstract**
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	Establishes the process used to control and authorize changes to engineering controlled documents to: Assure that total impact is considered before a change is approved and that the affected documents are identified and changed as approved Provide authority for a change and identify all pertinent interfaces and organizations responsible for these interfaces Provide accurate and traceable records of a change
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.

GE Hitachi Nuclear Energy Procedures and Policies		
Reference Number	**Document Title**	**Abstract**
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	Defines the requirements used to identify structures, systems, components, parts, and technical services that are safety-related. Safety-related structures, systems, components, and parts provide safety-related functions necessary to assure: The integrity of the reactor coolant pressure boundary; or The capability to shut down the reactor and maintain it in a safe shutdown condition; or 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.

GE Hitachi Nuclear Energy Procedures and Policies		
Reference Number	**Document Title**	**Abstract**
2.u	Commercial Grade Dedication of Software and Digital Components with Embedded Software for Use in Safety-Related Instrumentation and Control Applications, in Accordance with EPRI-TR-106439-1996	Establishes the requirements and responsibilities for dedicating commercial grade off the shelf software designated for use in safety-related applications.
2.v	Corrective Action Process	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.
2.w	Control of Nonconforming Material	Describes the methods by which nonconforming material is documented and controlled at GEH.
2.x	Quality and Technical Training	Defines the roles and responsibilities to assure personnel proficiency in quality and technical related activities. The Quality and Technical Training program: Ensures personnel are trained and proficient in assigned quality and technical tasks. Documents qualifications for technical positions, including minimum education, experience, and any special training requirements. Records training assignments in a centralized controlled training database.
2.z	Qualification of Previously Developed Software	Implements process for certifying the qualification of Previously Developed Software

GE Hitachi Nuclear Energy Procedures and Policies		
Reference Number	**Document Title**	**Abstract**
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."

Reference Number	**Document Title**	**Document Number**
4.	Guidelines on Evaluation and Acceptance of Commercial Grade Digital Equipment in Nuclear Safety Applications	EPRI TR-106439

2.4 ADDITIONAL IEEE STANDARD GUIDANCE

The following IEEE Standards provide additional guidance for the implementation activities. Conformance of this 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 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology
- 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 meets specified requirements for digital computer-based plant process control and monitoring software. The use of this SQAP 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 conformed to development standards.
- That test results adhere to acceptance standards in Section 7.0.

3.2 MANAGEMENT ORGANIZATION

3.2.1 Organization

This subsection defines the functional responsibilities and authorities of the Project organizations that are responsible for the quality of the software products.

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:

- Independent Verification and Validation (IV&V) of design documentation and outputs for Class Q software specified in the SMPM [2.3(1.a)]
- Verification and Validation (V&V) of design documentation and outputs for Class N3 and N2 software specified in the SMPM [2.3(1.a)]
- Safety analysis of Software Class Q software and Software Class N3 software

- Software and system testing
- Baseline reviews
- Configuration control
- Audits

Table 9-2 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 SQA Procedures and experienced 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 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.

Figure 2. (Deleted)

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 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 New Units Engineering. 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
- Ensuring independence of the SQA and SPE Organizations from the Design Organization.

The NPP Quality Manager shall either reserve authority, or shall formally designate a Quality Control Engineer (QCE) 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 and quality initiatives for improvement of processes for product and service offerings including:

- 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 Design Engineering Manager

The I&C Design Engineering organization, hereafter referred to as the Design Team, is described in the SMPM [2.3(1.a)]. The I&C 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.a)] and the required GEH procedures and policies. This includes the approval 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 SQAP (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 Class Q)

- Software Safety Team (SST, Software Class Q and N3)
- Baseline Review Team (BRT, Software Class Q, N3 and N2)

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 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 members
- Managing the conduct of IV&V tasks, and reviewing and approving the IV&V reports prepared by IVVT 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 responsible for determining the Software Class (described in Section 1.5, Software Classification) of the software and performing SSA. The SST shall coordinate with the IVVT to evaluate the V&V 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 the SSA for Software Class Q and N3 software
- Assigning tasks to SST members
- Managing the conduct of SST tasks and approving SSA reports prepared by SST members
- Ensuring that SSA is performed in accordance with the SSP described in Section 4.0

3.2.3.5 Configuration Management Manager

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

- Design Requirements and Inputs

- Design Activities
- Design Outputs
- Change Requests

3.2.3.6 Design Team

The Design Team is responsible for the design and implementation of the software products and the verification and validation of Software Class N3 and N2 software products. (The IVVT is responsible for the verification and validation of Software Class Q software products, and their independent verification and validation is described in section 5.) The responsible verifiers and testers shall be individual(s) or a group of individuals who are competent to perform verification and validation based on knowledge and experience. The V&V of Software Class N3 and N2 software products shall be conducted by qualified individual(s) or a group of individuals other than those who performed the design and development of the software product. These individuals may be from the Design organization, independent Software Project Engineering (SPE) or another independent third party which meets the requirements to perform the activity.

The independence criteria for V&V of Software Class N3 and N2 software products are defined in Independent Design Verification [2.3(2.i)]. The roles and responsibilities of the Design Team are described in the SMPM [2.3(1.a)].

3.2.4 Organizational Interfaces

The NPP Project Managers (PMs) are responsible for the commercial aspects of the software project. The detailed responsibility of the PM is described in the SMPM [2.3(1.a)].

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.a)].

3.2.5 Scheduling and Planning

The SPE and SQA Managers have the overall responsibility for scheduling and planning the tasks and activities described in this SQAPM. The Task Lead for each team (SST, IVVT, and 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 Work Planning and Scheduling [2.3(2.a)]. Each Task Lead is responsible for the preparation of a task-specific PWP.

While a cross-functional team performs the quality tasks, 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 quality tasks

- 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.a)] 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.a)].

Tables 9-1a through 9-1g present the required design and quality tasks and the associated task outputs for each software life cycle.

3.4 STANDARDS, PRACTICES, CONVENTIONS AND METRICS

3.4.1 Standards, Practices and Conventions

The applicable EOPs and P&Ps used in guiding the design and development of software products are specified in Section 2.3, Supplemental Documents. If detailed instructions are needed, project or platform/product line specific work practice instructions, such as ESBWR Project Instructions Engineering Service Instructions, or Work Instructions 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.a)], which at a minimum, shall include:

- Documentation standards
- Logic structure standards
- Coding standards
- Commentary standards
- Secure coding practices

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 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 conforms to its specifications
- The design adheres to regulations, standards, guidelines, plans, and procedures applicable to the project
- The design 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.

An individual other than the design document's Responsible Engineer (RE) shall conduct peer review. 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).

Design review is performed in accordance with GEH Design Review [2.3(2.e)]. For Software Class Q design documents, an SPE member shall participate in the design review. The review results shall be documented in a design review report as described in Subsection 5.4.2, Design Review Report.

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 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 to the SQA program and re-define the quality objectives. In addition to the review inputs required by Procedure 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]. The report 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:

- Close any Corrective Action Requests (CARs) associated with the project and project Design Record File (DRF)
- Set up warranty administration and review
- Conduct 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 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 confirm that the appropriate CI, which includes 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). 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, V&V, SSA, CySA and BR activities.

Nuclear Energy Quality Assurance Audit Requirements [2.3(3.d)], establishes the requirements and processes for a comprehensive audit program. This program confirms implementation of and compliance with the GEH quality system. It also determines 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 Corrective Action Process [(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 9-3 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 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 processes described in this SQAPM, the SMPM [2.3(1.a)], the required EOPs, P&Ps, and Corporate QA program be adhered to. Failure to comply with these processes 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
- Complaint from licensees
- Findings from regulatory authorities
- Other external organizations (e.g., International Standards Organization (ISO)/American Society of Mechanical Engineers (ASME) Code authorities)

[[

]]

3.7 TOOLS, TECHNIQUES AND METHODOLOGIES

3.7.1 Tools

The SPE and Design Team organizations shall employ the use of tools as needed to execute the tasks specified in this plan. If V&V is performed on the output produced by the tools each time the tools are used, then the tools used in part to perform V&V tasks do not need to be qualified.

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, V&V, 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, V&V and independent verification should include acceptance criteria for the design output. The NUREG 0800, SRP [2.2.1] 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, un-ambiguity, 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.

The traceability matrix shall clearly show the linkage between each requirement imposed on the software by the input documents. The matrix shall allow traceability in both directions. It shall be updated at the completion of each software life cycle phase. The final matrix shall permit tracing from the system requirements through the software requirements, design, implementation, testing, and installation. Figure 3 shows an example of a traceability matrix structure.

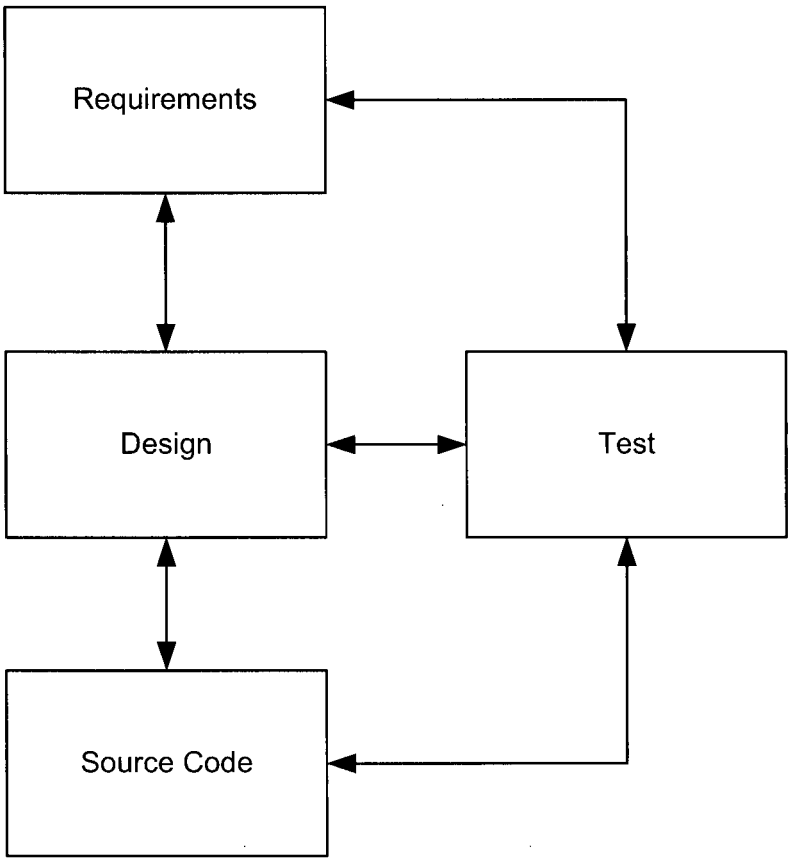

Figure 3. Example of A Traceability Matrix Structure

3.7.1.4 Product Data Management System

The 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 within the system. Roles, responsibilities and procedures are defined within PDMS [2.3(2.b)].

PDMS is the GEH-official CMS for engineering and quality controlled documents. Internal and external vendors providing the software products are not required to utilize PDMS [2.3(2.b)]. However, an appropriate computer-based CMS shall be used.

3.7.1.5 Design Record File

A Design Record File (DRF) is the 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 Design Record File, [2.3(2.g)].

3.7.1.6 Discrepancy Tracking System

A Discrepancy Tracking System (DTS) will be initiated to manage and track the anomalies identified during software testing. The software tests included will be the Software Validation Test (SVT), System Factory Acceptance Test (SFAT), [[
]] and Site Acceptance Test (SAT) (if SAT is within GEH scope of responsibility). Information that will allow the RE to evaluate the anomaly shall be included in the Discrepancy Tracking System. This information includes:

- 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 and 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), SCMP (Section 6.0), and STP (Section 7.0).

3.8 CODE AND MEDIA CONTROL

The computer-based design outputs, such as software source code, Commercial Off-the-Shelf (COTS) software, support software, and 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. At 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 audit 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 the public. 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.a)] 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 SQAPM. The IVVT shall independently verify and validate the PDS evaluation report prepared by the Design Team for software intended for use in Class Q software. The SMPM [2.3(1.a)] describes the evaluation of PDS.

3.10 RECORDS COLLECTION, MAINTENANCE, AND RETENTION

Section 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

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 Quality and Technical Training [2.3(2.x)]. The Design Team and the SPE teams shall be trained, either by self-study or classroom, this SQAPM, the SMPM [2.3(1.a)], applicable tools required to support the design and quality tasks, and the referenced EOPs and P&Ps. The training records shall be maintained in the training database.

3.12 RISK MANAGEMENT

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

Risk Management shall be implemented in accordance with 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. 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 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]. This SSP meets the guidelines specified in Chapter 7 of NUREG 0800 SRP [2.2.1] and the requirements outlined in Section 4.4 of IEEE Std. 1228, "IEEE Standard for Software Safety Plans" [2.4].

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

Safety critical software is defined as any 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 organizational responsibilities 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 for the completion of SSA activities:

- Prepare the SSA plan
- Obtain and allocate resources to ensure effective implementation of the SSP 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 licensee
- Ensure that required records are retained to document the conduct of software safety activities
- Participate in audits of the implementation of the SSP
- Ensure training of safety personnel in methods, tools, and techniques used in software safety analyses
- Ensure that the deviations and discrepancies are identified, documented and dispositioned by the Design Team

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 [2.3 (1.a)] Section 5.3, Organization of Software Life Cycle Process.

4.2.4 Documentation Requirements

This Subsection of the SSP specifies the documentation to be prepared and its contents. The list below delineates the requirements for documentation of safety-critical software.

- Software Project Management. Documentation of how the software safety program is implemented, integrated, and managed with the software development activities is discussed in this SQAPM and in the SMPM [2.3(1.a)].
- Software Configuration Management. Information regarding the CM of the design documentation and design outputs (including safety-critical design documents and design outputs for Class Q and N3) is specified in Section 6.0, Software Configuration Management Plan.
- Software Quality Assurance. Information regarding the SQA of Software Class Q and N3 design documentation and design outputs is specified in Section 3.0 of this SQAPM.
- Software Safety Requirements. Specification of safety-critical requirements to be implemented by the software to avoid or control system hazards are documented in the requirements documentation as specified in the SMPM [2.3(1.a)], Section 5.7.
- Software Safety Design. The design descriptions for safety-critical elements are specified in the SMPM [2.3(1.a)], Section 5.8.
- 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.a)] Subsection 5.8.3.3, Software Coding Conventions and Guidelines Document.
- Test Documentation. Test planning, test design, test cases, test procedures, and test reports for the SVT, SFAT, [[ ]] and SAT are specified in Section 7.7, Test Documentation. Tests produced for testing and verification of safety-critical elements do not need to be separate or isolated from other tests and may be included as part of the other tests. Portions of tests related to safety-critical elements shall be identified as such.
- Software IV&V. Information regarding how software safety-critical requirements 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 requirements to the design specifications, software source code, and software safety test cases.
- Reporting Software IV&V Activities. Information documenting the results of software IV&V activities is specified in Section 5.4, Verification and Validation Reporting.

- Software User Documentation. Information that is significant to the safe installation, use, maintenance, and retirement of the software product is specified in the SMPM [2.3(1.a)] Section 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 Discrepancies. Deviations and discrepancies identified during performance of software safety activities shall be documented and dispositioned prior to proceeding in accordance with Problem Reporting and Corrective Actions 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 6.8, Records Collection, Maintenance and Retention. Software safety records include the following:

- 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 software product
- Results of audits performed on software safety program tasks.
- Results of tests conducted on any safety-critical software modules, components or system.
- A record of training provided to software safety program personnel.
- Results of any certifications performed.
- Results of deviations and discrepancies.

The Software Safety Analysis Report shall be prepared to document the following:

- Name and Description of the Software Evaluated.
- System.
- Software Classification.

- Purpose and Scope.
- Reference Inputs.
- Software Safety Analysis Body of Report.
  - 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 Change Analysis.

Results of Deviations and Discrepancies. Deviations and discrepancies identified during performance of software safety analyses shall be documented and dispositioned in accordance with Problem Reporting and Corrective Actions in Section 3.6 of this SQAPM.

This documentation and disposition shall include:

- Anomalies noted
- Conclusion
- Responsible engineer
- Approving authority

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

The Requirements Traceability Matrix (RTM) shall be utilized as the tracking system to ensure that hazards, their responsibility assignment, and their status can be tracked throughout the software life cycle through retirement.

4.2.6 Software Configuration Management Activities

Software Configuration Management for safety-critical software 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. Configuration Management includes control of safety-critical element related design documentation, source code, object code, data, development tools, environments (both hardware and software), and test documentation.

Applicability of Configuration Management provisions ensures that additional requirements necessary for safety-critical elements are met for the following:

- Software development tools
- Previously developed software
- Vendor provided software

- Subcontractor-developed software
- V&V and SSA tools

Safety critical software is defined as any software for Software Class Q and N3 software products.

4.2.7 Software Quality Assurance Activities

Software Quality Assurance activities, as described in this SQAPM, shall assure proper performance of 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 the SQAPM in accordance with Section 4.10.
- The technical recommendations resulting from software safety tasks are reviewed, considered by change control authority, and, where appropriate, implemented.
- The reviews and audits shall address software safety concerns, requirements, guidelines, and process certification.
- The conduct of the software safety program shall be monitored.

4.2.8 Software V&V

Software V&V activities are specified in SQAPM Section 5.0, Software Verification and Validation Plan. The results of each life cycle phase will be matched against the system safety requirements and system hazard analysis to ensure:

- System safety requirements have been satisfied within the software life cycle phases
- No additional hazards have been introduced by the work done during the software 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 as specified in the SMPM [2.3(1.a)]. 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 software tools, the following areas shall be addressed in the SMPM [2.3(1.a)]:

- 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

The SST shall confirm that the Software Support Tool Documentation Package addresses tool approval, upgrade installation to an approved tool, approved tool withdrawal and identification of limitation on tool use.

4.2.10 Previously Developed or Purchased Software

Previously developed software (PDS) or commercial off-the-shelf (COTS) software that 1) can be qualified, 2) presents acceptable risk, or 3) remains safe in the context of its planned use shall be used in safety-critical software products. The inability to determine the level of risk present or the consequence of failure shall be sufficient justification for rejecting the use of the PDS or COTS software.

The SST shall evaluate the PDS or COTS Software Evaluation Report to:

- Verify the interfaces to and functionality of the PDS or COTS software that will be used in safety-critical systems
- Confirm the relevant documents that are available and determine their status
- Verify the PDS or COTS software to published specifications
- Verify the capabilities and limitations of the PDS or COTS software with respect to the project requirements are identified
- Verify the safety-critical features of the PDS or COTS software are validated using an appropriate test plan, design, cases and procedures
- Verify that a risk assessment has been conducted in the use of the PDS or COTS software and these will not result in an unacceptable level of risk

The results of this approval process shall be documented in the SSA Report. The inability to determine the level of risk present or the consequence of failure shall be sufficient justification for rejecting the use of PDS or COTS software.

4.2.11 Subcontract Management

Management of subcontractors for safety-related software shall be carried out in accordance with Vendor Control (Subsection 3.9.1). Design activities performed by a subcontractor shall be performed either in accordance with the SMPM 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 the affected organizations.

4.2.12 Process Certification

Process certification, performed to certify that the software product is produced in accordance with the SSP, is achieved through baseline reviews. Baseline reviews are described in the SVVP (SQAPM 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
- No additional hazards have been introduced by the work done
- 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

4.3.1 Software Safety Preparation Analysis

Software safety analysis preparation is performed by the SST during the Planning Phase of the software life cycle to: [[

]]

4.3.2 Software Safety Requirements Analysis

Software safety requirements analysis is performed during the Requirements Phase of the software life cycle to evaluate potential errors and deficiencies in the requirements that may contribute to a hazard.

4.3.2.1 SSA Inputs, Tasks and Outputs

SSA Inputs: [[

]]

Table 4.3.2.1-1 Software Safety Requirements Analysis Tasks and Outputs

SSA Tasks	SSA Outputs
1. [[	

SSA Tasks	SSA Outputs
	]]

4.3.3 Software Safety Design Analysis

Software safety design analysis is performed during the Design Phase of the software life cycle to confirm that the safety-critical portion of the software design correctly implements the safety-critical software functional requirements identified during the Requirements Phase, and the software design introduces no new hazards.

4.3.3.1 SSA Inputs, Tasks and Outputs

SSA Inputs: [[

]]

Table 4.3.3.1-1 Software Safety Design Analysis Tasks and Outputs

SSA Tasks	SSA Outputs
1. [[	

SSA Tasks	SSA Outputs

SSA Tasks	SSA Outputs

SSA Tasks	SSA Outputs
	]]

4.3.4 Software Safety Code Analysis

Software safety code analysis is performed during the Implementation Phase of the software life cycle to confirm that the safety-critical portions of the software design are correctly implemented in the software code, and the software coding introduces no new hazards.

4.3.4.1 SSA Inputs, Tasks and Outputs

SSA Inputs: [[

]]

Table 4.3.4.1-1 Software Safety Code Analysis Tasks and Outputs

SSA Tasks	SSA Outputs
1. [[	

SSA Tasks	SSA Outputs

SSA Tasks	SSA Outputs
	]]

4.3.5 Software Safety Test Analysis

Software safety test analysis is performed to confirm that the safety-critical portions of the software design are correctly implemented in the software code, and the software coding introduces no new hazards.

4.3.5.1 SSA Inputs, Tasks and Outputs

SSA Inputs: [[

]]

Table 4.3.5.1-1 Software Safety Test Analysis Tasks and Outputs

SSA Tasks	SSA Outputs
1. [[	

SSA Tasks	SSA Outputs

SSA Tasks	SSA Outputs

SSA Tasks	SSA Outputs

SSA Tasks	SSA Outputs

SSA Tasks	SSA Outputs

SSA Tasks	SSA Outputs

SSA Tasks	SSA Outputs
	]]

4.3.6 Software Safety Installation Analysis

Software safety installation analysis is performed during the Installation Phase of the software life cycle to confirm safety-critical requirements have been correctly implemented and the software functions safely within its specified environment.

4.3.6.1 SSA Inputs, Tasks and Outputs

SSA Inputs: [[

]]

Table 4.3.6.1-1 Software Safety Test Analysis Tasks and Outputs

SSA Tasks	SSA Outputs
1. [[	

SSA Tasks	SSA Outputs
	]]

4.3.7 Software Safety Change Analysis

Software safety change analysis is performed as part of Baseline Change Assessment during the Operations and Maintenance (O&M) Phase of the software life cycle to identify the safety-critical design elements that are affected directly or indirectly by the change request.

4.3.7.1 SSA Inputs, Tasks and Outputs

SSA Inputs: [[

]]

Table 4.3.7.1-1 Software Safety Change Analysis Tasks and Outputs

SSA Tasks	SSA Outputs
1. [[	

SSA Tasks	SSA Outputs

SSA Tasks	SSA Outputs
	]]

4.4 POST DEVELOPMENT

This section of the SSP defines the requirements for training, deployment, monitoring, maintenance, and retirement of safety-critical software that are necessary to ensure the continued safety of the system after its deployment and until its orderly retirement.

4.4.1 TRAINING

Training shall be provided in accordance with the systematic approach to training to assure safe operation of the software product. The Software Training Plan (STrngP) is described in Section 9.0 of the SMPM [2.3(1.a)].

4.4.2 DEPLOYMENT

4.4.2.1 Installation

Installation safety analysis tasks are described in Subsection 4.3.6. Compliance with these referenced sections assures installation of the software safety product consistent with the results of the software safety analyses.

4.4.2.2 Startup and Transition

Prior to starting up the newly installed safety-critical software product, the anomaly report 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 Licensee test plan and procedure.

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

The Installation Plan shall address the requirements for safely starting the new software product and, if an old software product is to be replaced, for making a safe transition from the old software product to the new software product. At a minimum, the following shall be assessed for applicability and where applicable implemented:

- Fallback modes for the new software product
- Startup of backup components and subsystems
- Startup of the new software product
- Parallel operation with backups
- Parallel operation of the old software product and the new software product
- Subsystem vs. full system operation
- Switchover to full system operation
- Validation of results from the new software product
- Cross validation of results between the old software product and the new software product
- Fallback in the case of failure of the new software product, including fallback to an old software product if one exists

4.4.2.3 Operations Support

The Operation and Maintenance (O&M) Manual shall be provided for the software product. The O&M Manual is described in Section 8.0 of the SMPM [2.3(1.a)].

4.4.3 MONITORING

The Licensee is responsible for monitoring the operation of the software product. Safety concerns that are detected during operation shall be documented and reported with the plant's problem-reporting procedures.

If GEH identifies a condition that could have potential safety implications for the software product, the Licensee shall be notified in accordance with applicable procedures.

4.4.4 MAINTENANCE

Software maintenance is discussed in Subsection 5.12.3 of the SMPM [2.3(1.a)].

4.4.5 RETIREMENT AND NOTIFICATION

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

4.5 PLAN APPROVAL

The Software Safety Plan is approved as part of this SQAPM in accordance with Section 3.0.

5. SOFTWARE VERIFICATION AND VALIDATION PLAN

This Software Verification and Validation Plan (SVVP) establishes the V&V tasks for the design and development of the 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 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.

V&V activity is limited to software designed and developed by GEH and GEH vendors. The list of selected software to undergo V&V shall be delineated in the PWP for the software project. Qualification of COTS software is performed by the Design Team as described in Subsection 5.8.3.6 of the SMPM [2.3(1.a)].

5.2 V&V OVERVIEW

5.2.1 Organization

Section 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

The software integrity level scheme approach is described in Section 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 Design Team is responsible for the Software Class N3 and N2 V&V activities.

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 V&V activities. The roles and responsibilities of the Design Team are described in the SMPM [2.3(1.a)]. 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 9-2 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 documentation and design outputs for a given software life cycle phase fulfilled (i.e., is traceable to) the requirements established in the previous phase. This is also performed to determine if the design documentation is 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 source code implements the specified design and does so in a manner that is compliant with the guidelines outlined in the applicable Software Coding Conventions and Guidelines Document. Code review shall be performed by a qualified software engineer other than the individual who performed the code implementation.

[[

]]

5.2.6.2.3 Software Functional Test

The software functional test includes the software module 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 System Requirements Specification (SyRS).

[[

]]

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 Audits.

[[

]]

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 Software Life Cycle Phase.

[[

]]

5.3.1 Management of V&V Activities

[[

]]

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

The following subsection describes the Management of V&V tasks:

5.3.1.1 SVVP Generation

An SVVP is generated as a stand-alone document or incorporated as part of the PWP for the software project in order to meet codes and standards, and in order to identify milestones, schedules and tasks for the software project. SVVP Generation Activities and Outputs are defined in Table 5.3.1.1-1.

V&V Inputs: [[

]]

Table 5.3.1.1-1 SVVP Generation V&V Activities and Outputs

Activities	Outputs
[[	]]

5.3.1.2 Baseline Change Assessment

Baseline changes assessment is described in Subsection 5.5.2, Baseline Change Assessment and Task Iteration Policy.

5.3.1.3 V&V Management Review

A management review of V&V is performed to ensure:

- Correct and timely implementation of the SVVP
- Effectiveness of the V&V effort in preventing and mitigating problems

- Compliance of the V&V effort with the requirements in the SVVP
- Basis of V&V
- Acceptance and certification of the software product

V&V Management Review Activities and Outputs are defined in Table 5.3.1.3-1.

V&V Inputs: [[

]]

Table 5.3.1.3-1 V&V Management Review Activities and Outputs

Activities	Outputs
[[	]]

5.3.1.4 Management and Technical Review Support

The IVVT shall provide support for management and technical reviews, including attendance at the design review and verification of timely completion of each V&V task in accordance with the schedule in the SVVP or PWP.. Management Technical Review Support Activities and Outputs are defined in Table 5.3.1.4-1.

Review Inputs: [[

]]

Table 5.3.1.4-1 Management and Technical Review Support Activities and Outputs

Activities	Outputs
[[ ]]	

5.3.1.5 Organizational Interfaces & Supporting Processes

The IVVT Lead shall provide organizational interfacing and support for coordination of V&V tasks with the SST, and to support SQA audits of the supplier. Organizational Interfaces and Supporting Processes Activities and Outputs are defined in Table 5.3.1.5-1.

V&V Inputs: [[ ]]

Table 5.3.1.5-1 Organizational Interfaces & Supporting Processes Activities and Outputs

Activities	Outputs
[[ ]]	

5.3.1.6 Acquisition Support V&V Activities (Acquisition Process)

The IVVT Task Lead shall provide IV&V support for the scoping, interface planning and requirements review of acquired software. Acquisition Support V&V Activities and Outputs are defined in Table 5.3.1.6-1.

V&V Inputs: [[ ]]

Table 5.3.1.6-1 Acquisition Support V&V Activities (Acquisition Process) and Outputs

Activities	Outputs
[[	

Activities	Outputs
	]]
NOTE: for Class N3 and N2 software, the activities delineated in steps 1 through 3 above are performed by Design Engineering I&C in accordance with section 3.9 of this manual.	

5.3.2 Planning Phase V&V Activities

The following subsections describe the V&V tasks to be conducted during the Planning Phase. The organizations responsible for these tasks are specified in Table 9-2.

5.3.2.1 Concept Documentation Evaluation

Concept documentation and system requirements are evaluated in the Planning Phase for the software project. Concept Documentation Evaluation Activities and Outputs are defined in Table 5.3.2.1-1.

Evaluation Inputs: [[

]]

Table 5.3.2.1-1 Concept Documentation Evaluation Activities and Outputs

Activities	Outputs
[[	]]

5.3.2.2 Software Plans Evaluation

The software plans to be used to control the software project are evaluated in the Planning Phase to ensure that codes, standards and regulatory guidance are appropriately captured and dispositioned. Software Plans Evaluation Activities and Outputs are defined in Table 5.3.2.1-1.

Evaluation Inputs: [[

]]

Table 5.3.2.2-1 Software Plans Evaluation Activities and Outputs

Activities	Outputs
[[	]]

5.3.2.3 Criticality Analysis

Subsection 4.3.1.1 describes the tasks necessary to establish the classification and hazard analysis results within the Planning Phase. Review and update of the existing classification and hazard analysis results (criticality analysis) from any prior Criticality Task Report may cause previously assigned software integrity levels to be raised or lowered for a given software element (i.e., requirement, module, function, subsystem, other software partition). Verify that no inconsistent or undesired software integrity consequences are introduced by reviewing the revised software integrity levels.

5.3.2.4 Hardware, Software and User Requirements Allocation Analysis

The allocation of the hardware, software and user requirements specified in the SDS are analyzed in the Planning Phase for correctness, accuracy and completeness. Hardware, software and user requirements allocation analysis activities and outputs are defined in Table 5.3.2.4-1.

Hardware, Software and User Requirements Allocation Analysis Inputs: [[

]]

Table 5.3.2.4-1 Hardware, Software and User Requirements Allocation Analysis Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.2.5 Traceability Analysis

Requirements Traceability Analysis (RTA) is initiated in the Planning Phase to identify the system requirements (contained in the SDS and Logic Diagrams) that will be implemented completely or partially by software, and that these requirements are traceable to the contract. The

RTA results will be placed into a requirements traceability matrix (RTM) and summarized in a RTM report. RTA activities and outputs are defined in Table 5.3.2.5-1.

RTA Inputs: [[

]]

Table 5.3.2.5-1 Traceability Analysis Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.2.6 [[ ]] and SAT Plan Generation and Evaluation

[[ ]] and SAT Plan shall be generated to describe the test scope, approach, resources, schedules, and risks requiring contingency planning of the testing activities to ensure the software correctly implements system and software requirements in an operational environment. The test plan shall be generated in accordance with the test document purpose, format, and content specified in Section 7.0. [[ ]] and SAT Plan verification activities and outputs are defined in Table 5.3.2.6-1.

V&V Inputs: [[

]]

Table 5.3.2.6-1 [[ ]] and SAT Plan Generation Evaluation Activities and Outputs

Activities	Outputs
[[ ]]	

5.3.2.7 Hazard Analysis

Hazard analysis is performed on the concept documents to identify the potential hazards to and from the conceptual system. Hazard analysis activities and outputs are defined in Table 5.3.2.7-1.

Hazard Analysis Inputs: [[ ]]

Table 5.3.2.7-1 Hazard Analysis Activities and Outputs

Activities	Outputs
[[ ]]	

5.3.2.8 Risk Analysis

Review the hazard analysis report, contract and PWP to assess the technical and management risks and provide recommendations to eliminate, reduce, or mitigate the risks. Risk analysis activities and outputs are defined in Table 5.3.2.8-1.

Risk Analysis Inputs: [[

]]

Table 5.3.2.8-1 Risk Analysis Activities and Outputs

Activities	Outputs
[[	]]

5.3.2.9 Acquisition V&V Activities

The acquisition V&V activities during the Planning Phase addresses the initiation, preparation of response, contract, planning, execution and control, review and evaluation, and delivery and completion activities. Acquisition V&V activities for the Planning Phase and outputs are defined in Table 5.3.2.9-1.

Table 5.3.2.9-1 Acquisition V&V Activities and Outputs

Activities	Outputs
[[	

Activities	Outputs
	]]

5.3.2.10 Configuration Management Assessment

The Configuration Management Assessment activities during the Planning Phase address baseline review of the scope of the software product, the verified design outputs, nuclear safety, technical and contractual requirements, and the placement of the SQA process. Configuration Management Assessment activities and outputs are defined in Table 5.3.2.10-1.

Configuration Management Assessment Inputs: [[

]]

Table 5.3.2.10-1 Configuration Management Assessment Activities and Outputs

Activities	Outputs
[[	]]

5.3.3 Requirements Phase V&V Activities

The following subsections describe the Activities to be conducted during the Requirements Phase. The organizations responsible for these tasks are specified in Table 9-2.

5.3.3.1 Traceability Analysis

Requirements Traceability Analysis (RTA) traces the software requirements [contained in the SRS, SyRS, Data Communication Protocol Specification (DCPS), and User Interface Specification (UIS)] to system requirements (contained in the SDS and HSS), and system requirements to the software requirements. The RTA analyzes the identified relationships for correctness, consistency, completeness, and accuracy. The RTA results will be placed into a requirements traceability matrix (RTM) and summarized in a RTM report. RTA activities and outputs are defined in Table 5.3.3.1-1.

RTA Inputs: [[

]]

Table 5.3.3.1-1 Requirements Traceability Analysis Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.3.2 Software Requirements Evaluation

Evaluate the requirements (e.g., functional, capability, interface, qualification, safety, security, data definitions, installation, and acceptance) of the HSS, SRS, SyRS and DCPS for correctness,

consistency, completeness, accuracy, readability, and testability. Requirements related to human factors, user documentation, user operation, and user maintenance are evaluated separately, see Subsection 5.3.4.3. User Interface Specification Evaluation. Software Requirements Evaluation activities and outputs are defined in Table 5.3.3.2-1.

Software Requirements Evaluation Inputs: [[

]]

[[

]]

Table 5.3.3.2-1 Software Requirements Evaluation Activities and Outputs

Activities	Outputs
1. [[	]]

Activities	Outputs

Activities	Outputs
]]	

5.3.3.3 User Interface Specification Evaluation

Evaluate the requirements (e.g., human factors, user documentation, user operation, and user maintenance) of the User Interface Specification (UIS) for correctness, consistency, completeness, accuracy, readability, and testability. UIS Evaluation activities and outputs are defined in Table 5.3.3.3-1.

UIS Evaluation Inputs: [[

]]

Table 5.3.3.3-1 UIS Evaluation Activities and Outputs

Activities	Outputs
1. [[	

Activities	Outputs

Activities	Outputs
]]	

5.3.3.4 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. Interface Analysis activities and outputs are defined in Table 5.3.3.4-1.

Interface Analysis Inputs: [[

]]

Table 5.3.3.4-1 Interface Analysis Activities and Outputs

Activities	Outputs
1. [[	

Activities	Outputs

]]

5.3.3.5 Analysis Tasks and Documents

Subsection 4.3.2.1 describes the criticality analysis and the hazards analysis tasks for the Requirements Phase.

5.3.3.6 Risk Analysis

Review and update risk analysis using prior task reports. Provide recommendations to eliminate, reduce, or mitigate the risks. Requirements Risk Analysis Report is file with the Management Review Report. Risk analysis activities and outputs are defined in Table 5.3.3.6-1.

Analysis Inputs: [[

]]

Table 5.3.3.6-1 Risk Analysis Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.3.7 Configuration Management Assessment

A configuration management assessment is performed to confirm that the Configuration Management process is complete and adequate. Configuration management assessment activities and outputs are defined in Table 5.3.3.7-1.

Configuration management assessment Inputs: [[

]]

Table 5.3.3.7-1 Configuration Management Assessment Activities and Outputs

Activities	Outputs
[[	

]]

5.3.4 Design Phase V&V Activities

The following subsections describe the V&V tasks to be conducted during the Design Phase. Organizations responsible for these tasks are specified in Table 9-2.

5.3.4.1 Traceability Analysis

Trace design elements (SDD, SVT Design, SVT Cases, SVT Procedures, SFAT Design, SFAT Cases, SFAT Procedures, and SFAT Specifications) to requirements [SyRS, SRS, DCPS, UIS, Intra-System Communication Protocol Specification (ISCPS)], and requirements to design elements. Analyze relationships for correctness, consistency, and completeness. Design traceability analysis activities and outputs are defined in Table 5.3.4.1-1.

Design Traceability Analysis Inputs: [[

]]

Table 5.3.4.1-1 Traceability Analysis Activities and Outputs

Activities	Outputs
[[	

]]

Activities	Outputs

]]

5.3.4.2 Software Design Evaluation

Evaluate the software design evaluation inputs for correctness, consistency, completeness, accuracy, readability, and testability. Software design evaluation activities and outputs are defined in Table 5.3.4.2-1.

Software Design Evaluation Inputs: [[

]]

Table 5.3.4.2-1 Software Design Evaluation Activities and Outputs

Activities	Outputs
[[	

Activities	Outputs

]]

5.3.4.3 Test Plans Generation and Verification

Plan V&V testing to validate that the software system correctly implements system requirements. The task criteria are 1) compliance with design requirements; 2) assessment of timing, sizing, and accuracy; 3) performance at boundaries and interfaces and under stress and error conditions; and 4) measures of requirements test coverage and software reliability and maintainability. Test Plans Generation and Verification activities and outputs are defined in Table 5.3.4.3-1.

Test Plans Generation and Verification Inputs: [[

]]

Table 5.3.4.3-1 Test Plans Generation and Verification Activities and Outputs

Activities	Outputs
[[	]]

5.3.4.4 Interface Analysis

Verify and validate that the software design interfaces with hardware, user, operator, software, and other systems for correctness, consistency, completeness, accuracy, and testability. Interface analysis activities and outputs are defined in Table 5.3.4.4-1.

Interface Analysis Inputs: [[

]]

Table 5.3.4.4-1 Interface Analysis Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.4.5 Criticality Analysis

Subsection 4.3.3.1 describes the tasks necessary to confirm the classification and hazard analysis results from the Requirements Phase for the Design Phase. Review and update the existing classification and hazard analysis results (criticality analysis) results from the prior Criticality

Task Report using the SDD. Implementation methods and interfacing technologies may cause previously assigned software integrity levels to be raised or lowered for a given software element (i.e., requirement, module, function, subsystem, other software partition). Verify that no inconsistent or undesired software integrity consequences are introduced by reviewing the revised software integrity levels.

5.3.4.6 Risk Analysis

Review and update risk analysis using prior task reports. Risk analysis activities and outputs are defined in Table 5.3.4.6-1.

Risk analysis Inputs: [[

]]

Table 5.3.4.6-1 Risk Analysis Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.4.7 Configuration Management Assessment

The configuration management assessment is an audit of the software development process to ensure that the software development plans are being followed. The configuration management assessment is performed integral with the baseline review. The Configuration Management Assessment activities and outputs are defined in Table 5.3.4.7-1.

Configuration Management Assessment Inputs: [[

]]

Table 5.3.4.7-1 Configuration Management Assessment Activities and Outputs

Activities	Outputs
[[	]]

5.3.5 Implementation Phase V&V Activities

The following subsections describe the V&V tasks to be conducted during the Implementation Phase. The organizations responsible for these tasks are specified in Table 9-2.

5.3.5.1 Traceability Analysis

Requirements Traceability Analysis (RTA) maps the source code components to corresponding design specifications [contained in the SDS and ISCPS]. The RTA analyzes the identified relationships for correctness, consistency and completeness. The RTA results will be placed into a requirements traceability matrix (RTM) and summarized in a RTM report. RTA activities and outputs are defined in Table 5.3.5.1-1.

RTA Inputs: [[

]]

Table 5.3.5.1-1 Traceability Analysis Activities and Outputs

Activities	Outputs
[[	]]

5.3.5.2 Source Code Components Evaluation

Evaluate the source code components (Source Code Documentation) for correctness, consistency, completeness, accuracy, readability, and testability. The Source Code and Source Code Documentation Evaluation activities and outputs are defined in Table 5.3.5.2-1.

Evaluation Inputs: [[

]]

Table 5.3.5.2-1 Source Code and Source Code Documentation Evaluation Activities and Outputs

Activities	Outputs
[[	]]

Activities	Outputs

Activities	Outputs

]]

5.3.5.3 Interface Analysis

Verify and validate the correctness, consistency, completeness, accuracy, and testability of the interfaces of the software source code with hardware, user, operator, software, and other systems. The interface analysis activities and outputs are defined in Table 5.3.5.3-1.

Interface Analysis Inputs: [[

]]

Table 5.3.5.3-1 Interface Analysis Activities and Outputs

Activities	Outputs

Activities	Outputs
]]	

5.3.5.4 Criticality Analysis

Subsection 4.3.4.1 describes the tasks necessary to confirm the classification and hazard analysis results from the Design Phase for the Implementation Phase. Review and update the existing classification and hazard analysis results (criticality analysis) results from the prior Criticality Task Report using the source code. Implementation methods and interfacing technologies may cause previously assigned software integrity levels to be raised or lowered for a given software element (i.e., requirement, module, function, subsystem, other software partition). Verify that no inconsistent or undesired software integrity consequences are introduced during software implementation by reviewing the revised software classification level.

5.3.5.5 Software Functional Test Execution

Perform software functional testing, analyze test results to validate that software correctly implements the design. Document the results as required by the Test Execution Guidelines described in SMPM [2.3(1.a)], Subsection 6.11.1.3. Software functional test activities and outputs are defined in Table 5.3.5.5-1.

SFT Inputs: [[

]]

Table 5.3.5.5-1 Software Functional Test Execution and Verification Activities and Outputs

Activities	Outputs
[[	

Activities	Outputs

Activities	Outputs

Activities	Outputs
	]]

5.3.5.6 Software Functional Test Report Verification

The Software Functional Test Report shall be verified to assure the software function test is complete, correct, accurate, and test results are traceable to the requirements and design documents. The Software Functional Test Report Verification Activities and Outputs are defined in Table 5.3.5.6-1.

V&V Inputs: [[

]]

Table 5.3.5.6-1 Software Functional Test Report Verification Activities and Outputs

Activities	Outputs
1. [[	

Activities	Outputs

]]

5.3.5.7 Software Build Description Evaluation

The evaluation of the Software Build Description in the Implementation Phase assures that the software can be reproduced from modules and libraries without ambiguity. The Software Build Description Evaluation activities and outputs are defined in Table 5.3.5.7-1.

Evaluation Inputs: [[

]]

Table 5.3.5.7-1 Software Build Description Evaluation Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.5.8 Software Validation and SFAT Test Design, Test Cases and Test Procedure Specifications Evaluation

The Software Validation and SFAT Test Design, Test Cases and Test Procedure Specifications shall be verified to assure these test documents are complete, correct, accurate and traceable to the system requirements. The evaluation activities and outputs are defined in Table 5.3.5.8-1.

Evaluation Inputs: [[

]]

Table 5.3.5.8-1 Software Validation and SFAT Test Design, Test Cases and Test Procedure Specifications Evaluation Activities and Outputs

Activities	Outputs
1. [[	

Activities	Outputs
	]]

5.3.5.9 Risk Analysis

Review and update risk analysis using prior task reports. Risk analysis activities and outputs are defined in Table 5.3.5.9-1.

Risk analysis Inputs: [[

]]

Table 5.3.5.9-1 Risk Analysis Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.5.10 Configuration Management Assessment

The configuration management assessment is an audit of the software development process to ensure that the software development plans are being followed. The configuration management assessment is performed integral with the baseline review. Configuration Management Assessment Activities and Outputs are defined in Table 5.3.5.10-1.

Configuration Management Assessment Inputs: [[

]]

Table 5.3.5.10-1 Configuration Management Assessment Activities and Outputs

Activities	Outputs
[[	]]

5.3.6 Test Phase V&V Activities

The following subsections describe the V&V activities to be conducted during the Test Phase. The organizations responsible for these tasks are specified in Table 9-2.

5.3.6.1 Traceability Analysis

Trace [[ ]] test cases and procedures to requirements (SDS and Logic Diagrams), and requirements to [[ ]]. Trace Site Acceptance Test (SAT) test cases and procedures to requirements (SDS, Logic Diagrams and Contract), and requirements to SAT test cases and procedures. Analyze relationships for correctness, consistency, and completeness. Traceability analysis activities and outputs are defined in Table 5.3.6.1-1.

Traceability Analysis Inputs: [[

]]

Table 5.3.6.1-1 Traceability Analysis Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.6.2 Software Validation Test Execution

Perform Software Validation Test (SVT) using an approved test procedure to validate that the software is operational and conforms to the functional and performance requirements specified by the HSS, SyRS, SRS, UIS and DCPS. SVT activities and outputs are defined in Table 5.3.6.2-1.

Test Inputs: [[

]]

Table 5.3.6.2-1 Software Validation Test Execution Activities and Outputs

Activities	Outputs
1. [[	

Activities	Outputs
	]]

5.3.6.3 Software Validation Test Report Evaluation

The Software Validation Test Report shall be verified to assure the software validation test is complete, correct, accurate, and test results are traceable to the requirements documents. SVT Test Report evaluation activities and outputs are defined in Table 5.3.6.3-1.

Evaluation Inputs: [[

]]

Table 5.3.6.3-1 Software Validation Test Report Evaluation Activities and Outputs

Activities	Outputs
1. [[	

Activities	Outputs
	]]

5.3.6.4 [[ ]] *and SAT Test Design, Case and Procedure Specifications Evaluation*

Evaluate [[ ]] and SAT test documentation for correctness and completeness. The evaluation activities and outputs are defined in Table 5.3.6.4-1.

Evaluation Inputs: [[

]]

Table 5.3.6.4-1 [[ ]] and SAT Test Design, Case and Procedure Specifications Evaluation Activities and Outputs

Activities	Outputs
[[	]]

5.3.6.5 Hazard Analysis

Subsection 5.7.9 in SMPM [2.3(1.a)] described qualification of software support tool to ensure tool used to support testing will not inadvertently introduce new hazards. Hazard analysis shall be performed on test tool used to support testing of Class Q software. Hazard analysis activities and outputs are defined in Table 5.3.6.5-1.

Hazard Analysis Inputs: [[

]]

Table 5.3.6.5-1 Hazard Analysis Activities and Outputs

Activities	Outputs
[[	]]

5.3.6.6 Risk Analysis

Review and update risk analysis using prior task reports. Risk analysis activities and outputs are defined in Table 5.3.6.6-1.

Risk analysis Inputs: [[

]]

Table 5.3.6.6-1 Risk Analysis Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.6.7 Configuration Management Assessment

The configuration management assessment is an audit of the software development process to ensure that the software development plans are being followed. The configuration management assessment is performed integral with the baseline review. The configuration management assessment activities and outputs are defined in Table 5.3.6.7-1.

Configuration Management Assessment Inputs: [[

]]

Table 5.3.6.7-1 Configuration Management Assessment Activities and Outputs

Activities	Outputs
[[	]]

5.3.6.8 Software O&M and Software Training Manual Evaluation

Evaluate Software O&M Manual and Software Training Manual for correctness, accuracy, completeness and readability. The evaluation activities and outputs are defined in Table 5.3.6.8-1.

Evaluation Inputs: [[

]]

Table 5.3.6.8-1 Software O&M and Software Training Manual Evaluation Activities and Outputs

Activities	Outputs
[[	]]

5.3.7 Installation Phase V&V Activities

The following subsections describe the V&V tasks to be conducted during the Installation Phase. The organizations responsible for these tasks are specified in Table 9-2.

5.3.7.1 System Factory Acceptance Test Execution

Perform System Factory Acceptance Test (SFAT) using an approved test procedure to validate that the software product is operational, conformed to the functional and performance requirements specified by the SFAT Plan and performed as intended. SFAT activities and outputs are defined in Table 5.3.7.1-1.

SFAT Inputs: [[

]]

Table 5.3.7.1-1 System Factory Acceptance Test Execution Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.7.2 System Factory Acceptance Test Report Evaluation

The SFAT Report shall be verified to assure the software factory acceptance test is complete, correct, accurate, and test results are traceable to the requirements documents. SFAT Report evaluation activities and outputs are defined in Table 5.3.7.2-1.

Evaluation Inputs: [[

]]

Table 5.3.7.2-1 System Factory Acceptance Test Report Evaluation Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.7.3 [[ ]] ***and Site Acceptance Test Execution***

[[

]]

[[ ]] and SAT Inputs: [[

]]

Table 5.3.7.3-1 [[ ]] and Site Acceptance Test Execution Activities and Outputs

Activities	Outputs
1. [[	

]]

5.3.7.4 [[ ]] *and Site Acceptance Test Report Evaluation*

[[

]]

Evaluation Inputs: [[

]]

Table 5.3.7.4-1 [[ ]] and Site Acceptance Report Evaluation Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.7.5 Installation Configuration Audit

Verify that the 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. Installation configuration audit activities and outputs are defined in Table 5.3.7.5-1.

Installation Configuration Audit Inputs: [[

]]

Table 5.3.7.5-1 Installation Configuration Audit Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.7.6 Installation Configuration Table Generation and Evaluation

Generate in accordance with the SMPM [2.3(1.b)], subsection 7.5.3, and evaluate the Installation Configuration Table for completeness, correctness, consistency, accuracy and traceability. Evaluation activities and outputs are defined in Table 5.3.7.6-1.

Evaluation Inputs: [[

]]

Table 5.3.7.6-1 Installation Configuration Table Generation and Evaluation Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.7.7 Installation Checkout

Verify that the software and software products have been correctly installed and operate in the target environment. Installation checkout activities and outputs are defined in Table 5.3.7.7-1.

Installation Checkout Inputs: [[

]]

Table 5.3.7.7-1 Installation Checkout Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.7.8 Hazard Analysis and Risk Analysis

Review and update hazard and risk analysis using prior task reports. Hazard and risk analysis activities and outputs are defined in Table 5.3.7.8-1.

Risk Analysis Inputs: [[

]]

Table 5.3.7.8-1 Hazard and Risk Analysis Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.7.9 Configuration Management Assessment

The configuration management assessment is an audit of the software development process to ensure that the software development plans are being followed. The configuration management assessment is performed integral with the baseline review. Configuration Management Assessment activities and outputs are defined in Table 5.3.7.9-1.

Configuration Management Assessment Inputs: [[

]]

Table 5.3.7.9-1 Configuration Management Assessment Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.7.10 V&V Final Report Generation

V&V activities and results shall be assessed at the end of the Installation Phase or at the conclusion of the V&V effort. Assessment activities and outputs are defined in Table 5.3.7.10-1.

V&V Inputs: [[

]]

Table 5.3.7.10-1 V&V Final Report Generation Activities and Outputs

Activities	Outputs
[[	]]

5.3.8 Operations and Maintenance Phase V&V Activities

The following subsections describe the V&V tasks to be conducted during the Operations and Maintenance Phase. The organizations responsible for these tasks are specified in Table 9-2.

5.3.8.1 Evaluation of New Constraints

New Constraints evaluation is performed as part of Baseline Change Assessment. Baseline Change Assessment is described in Subsection 5.5.2. New constraints evaluation activities and outputs are defined in Table 5.3.8.1-1.

Evaluation Inputs: [[

]]

Table 5.3.8.1-1 Evaluation of New Constraints Activities and Outputs

Activities	Outputs
1. [[ ]]	

5.3.8.2 Proposed Change Assessment

Proposed change assessment is performed as part of Baseline Change Assessment. Baseline Change Assessment is described in Subsection 5.5.2. Proposed change assessment activities and outputs are defined in Table 5.3.8.2-1.

Proposed Change Assessment Inputs: [[ ]]

Table 5.3.8.2-1 Proposed Change Assessment Activities and Outputs

Activities	Outputs
1. [[ ]]	

5.3.8.3 SVVP Revision

Revise SVVP to incorporate new requirements needed to modify software product. SVVP revision activities and outputs are defined in Table 5.3.8.3-1.

SVVP Revision Inputs: [[

]]

Table 5.3.8.3-1 SVVP Revision Activities and Outputs

Activities	Outputs
[[ ]]	

5.3.8.4 Anomaly Evaluation

Anomalies identified during the operation of the software product are documented within the Baseline Change Assessment. Baseline Change Assessment is described in Subsection 5.5.2. Proposed change assessment activities and outputs are defined in Table 5.3.8.4-1.

Anomaly Evaluation Inputs: [[ ]]

Table 5.3.8.4-1 Anomaly Evaluation Activities and Outputs

Activities	Outputs
1. [[ ]]	

5.3.8.5 Regression Analysis

Regression analysis is performed to identify the extent to which the regression test is to be performed as a consequence of modifications. Regression analysis activities and outputs are defined in Table 5.3.8.5-1.

Regression Analysis Inputs: [[

]]

Table 5.3.8.5-1 Regression Analysis Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.8.6 Migration Assessment

Assess the software requirements and implementation to ensure the following topics are addressed:

- Migration requirements
- Migration tools
- Conversion of software products and data
- Security requirements
- Software archiving
- Support for the prior environment
- User notification

Migration assessment activities and outputs are defined in Table 5.3.8.6-1.

Migration Analysis Inputs: [[

]]

Table 5.3.8.6-1 Migration Assessment Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.8.7 Retirement Assessment

Assess the installation package to ensure the following topics are addressed: [[

]]

Retirement assessment activities and outputs are defined in Table 5.3.8.7-1.

Retirement Analysis Inputs: [[

]]

Table 5.3.8.7-1 Retirement Assessment Activities and Outputs

Activities	Outputs
1. [[ ]]	

5.3.8.8 Hazard Analysis and Risk Analysis

Subsection 4.3.7 describes the safety change analysis for the Operations and Maintenance (O&M) Phase.

Review and update hazard and risk analysis using prior task reports. Hazard and risk analysis activities and outputs are defined in Table 5.3.8.8-1.

Subsection 4.3.4.1 addresses the tasks necessary to verify that no inconsistent or unintentional functions are introduced during coding and that no new hazards are introduced.

Hazard and Risk Analysis Inputs: [[ ]]

Table 5.3.8.8-1 Hazard and Risk Analysis Activities and Outputs

Activities	Outputs
1. [[ ]]	

5.3.8.9 Task Iteration

Perform the V&V tasks identified during SVVP revision (see Subsection 5.3.8.3) to ensure:

- The planned changes are implemented correctly

- Documentation is complete and current
- Changes do not cause unacceptable or unintended system behaviors (see Subsection 5.3.8.5)

Iterated V&V activities and outputs are defined in Table 5.3.8.9-1.

V&V Inputs: [[

]]

Table 5.3.8.9-1 Task Iteration Activities and Outputs

Activities	Outputs
1. [[	]]

5.3.9 Acquired Software and Vendor V&V Tasks

Acquired software refers to:

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

5.3.9.1 Software Support Tools

[[

]]

5.3.9.2 Commercial Off The Shelf Software

COTS software is software that is commercially available to the public. 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(4)]. Commercial Grade Dedication of Software and Digital Components with Embedded Software for Use in Safety-Related Instrumentation and Control Applications, in Accordance with EPRI-TR-106439-1996 [2.3(2.u)] provides additional guidance in dedicating commercial off the shelf software designated for use in safety-related applications.

Applicable portions of Commercial Grade Dedication of Software and Digital Components with Embedded Software for Use in Safety-Related Instrumentation and Control Applications, in Accordance with EPRI-TR-106439-1996 may be used for guidance in evaluating commercial off the shelf software designated for use in nonsafety-related applications.

[[

]]

5.3.9.3 Previously Developed Software

[[

]]

5.3.9.4 Vendor Software

[[

]]

5.4 V&V REPORTING

V&V 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). The following Subsections describe the V&V reports that are required by this V&V process.

5.4.1 Independent Verification Package

[[

]]

5.4.2 Design Review Report

[[

]]

5.4.3 Test Report

The test report is described in Subsection 7.7.8.

5.4.4 Anomaly Report

[[

]]

5.4.5 Baseline Review Record

[[

]]

5.4.6 Managerial Review Report

[[

]]

5.4.7 V&V Final Report

[[

]]

5.5 V&V ADMINISTRATIVE REQUIREMENTS

The Verification and Validation Administrative Requirements used in conjunction with the V&V activities are outlined in the following subsections.

5.5.1 Anomaly Resolution and Reporting

[[

]]

5.5.2 Baseline Change Assessment and Task Iteration Policy

[[

]]

5.5.3 Deviation Policy

[[

]]

5.5.4 Control Procedures

The V&V activities records are configured, protected, and maintained in accordance with the procedures described in Section 6.0, Software Configuration Management Plan.

5.5.5 Standards, Practices, and Conventions

The standards, practices, and conventions that govern this SVVP are specified in Section 3.4. Additional standards, practices, and conventions for a project as required by contract shall be stated in the project-specific PWP and documented in the DRF.

5.5.6 Test Documentation Requirements

The purpose, format, and content of the test documents used to support the V&V are specified in Section 7.0, Software Test Plan.

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 [2.2.4].

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 change requests (as defined in 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 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 Engineering Change Control [2.3(2.p)].

The SCM responsibilities of the I&C Design Engineering Manager is to review and approve the initiation or change of design documents for Software Class Q, N3 and N2 software to confirm:

- V&V were performed by technically competent individual(s).
- Scope of review and verification per 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 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 the approved 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 verification in accordance with 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 Management 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, who 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 P&Ps, EOPs, and directives applicable to the SCM activities, are specified in Section 2.3. These policies and procedures are used to supplement the process specified in this SCMP. If any external constraints are placed on the software product per contract requirements, such constraints and its impact and effect on the SCMP shall be documented in the project-specific PWP.

6.2.4 SCM Schedule

The SCM schedule that establishes the sequence and SCM tasks shall be specified in the PWP by 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]) is a web-based database used to track:
  - 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 Design Review [2.3(2.e)]

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

6.3.2 SCM Techniques

Section 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.

  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 7.7.8), Audit Reports (Subsection 3.5.2) and SSA Reports (Subsection 4.2.5) 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. Design Record File [2.3(2.g)] defines the procedures to establish and maintain a DRF.

- Acquired software such as support software and 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 source file 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 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 RCCE.

Figure 5 shows an example of the naming convention.

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

  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.

  CIs shall be placed under configuration control and stored in the PDMS (Subsection 3.7.1.4). Table 9-4 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

Design control measures for the software product 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 V&V methods and implementation by individuals or groups not directly responsible for the original design.

The design process performed shall be composed of the activities defined by 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.a)] and Design Process [2.3(2.f)]. 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 Product Data Management System (PDMS) [2.3(2.b)] is used for this purpose. PDMS is described in Subsection 3.7.1.4. V&V 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 of I&C software. The purpose of this process is to:

- Ensure that the impact, risks and hazards are considered before a change is approved
- Ensure that the documents are identified and revised 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 or error with a software product, as described below. Reasons for a proposed change are categorized in Table 6.4.2.2-1.

Table 6.4.2.2-1 Reasons for Change Request

Life Cycle Phase	Reasons
Requirements, Design, Implementation	• Design requirements • Change in regulatory requirement or codes and standards requirement • Change request from Vendor (Subsection 3.9.1 Vendor Control)
Tests	• Anomaly or error correction during V&V and testing, • Change request from Vendor (Subsection 3.9.1 Vendor Control)
Installation	• Anomaly or error correction during software product installation

Life Cycle Phase	Reasons
Operation and Maintenance	• Anomaly or error correction during operation and maintenance of the software product, • Licensee contract change request

[[

]]

Table 6.4.2.2-2 Change Process Steps

Responsible Individual	Change Process Steps
[[	]]
[[	]]

Responsible Individual	Change Process Steps
[[	]]
[[	]]
[[	]]
[[	]]

6.4.2.3 Change Request During Requirements, Design, and Implementation Phase

[[

]]

6.4.2.4 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.

6.4.2.5 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 Field Deviation Disposition Request [2.3(2.q)].

6.4.2.6 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 via an issue tracking tool, for example, the Nuclear Customer Issue Resolution (CIR) Tool [2.3(2.y)]. If an error is found in either Software Class Q or Class N3 software, then Reporting of Defects and Noncompliance under 10 CFR Part 21 [2.3(3.e)] applies.

6.4.2.7 Change Request from Licensee

Proposed changes to a software product design due to contract revision shall be processed in accordance with Baseline Change Assessment and Task Iteration Policy (Subsection 5.5.2).

6.4.2.8 Change Notification

If discrepancies or errors affect software products already 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)

If an error is found in either Software Class Q or Class N3 software, then Reporting of Defects and Noncompliance under 10 CFR Part 21 [2.3(3.e)] applies.

6.4.2.9 Design Interfaces Control

Engineering design interfaces with vendors or design organizations supporting the design of the software product shall be formally controlled 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 Audit
- Physical Audit

6.4.4.1 Functional 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 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 items identified as being part of the configuration are present in the product baseline. The audit shall 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. The following baselines have been designated by the SMPM [2.3(1.a)]:

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

The SMPM [2.3(1.a)], 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
- CySA tasks performed adhere to the procedures outlined in the CySPP [2.3(1.b)]

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, SSA and CySA, 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 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, CySA, SSA, and V&V tasks of a particular software life cycle phase.
5. The BRT has the authority to approve 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 shall 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 shall be reviewed by the BRT to confirm that:

- Adherence to the SMPM, SQAPM and CySPP has been achieved
- The required documents have been completed and verified
- The V&V scope and approach is reasonable
- Comments made during the V&V 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 to release the 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 SCMP and the SMPM [2.3(1.a)].

The equivalent SCMP shall be submitted to GEH for review and approval. Supplier Design Services Document Review [2.3(2.c)] and 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.a)] 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 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. The 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 test activities associated with the software development process. The STP also identifies the items being tested, the features to be tested, the test tasks to be performed, the personnel responsible for each task, and the risks associated with this plan.

7.2 SCOPE

The STP applies to the digital I&C software. Applicability is not restricted by the size, complexity, or criticality of the software. This plan applies to digital I&C software classification levels. This plan applies to the phases of testing from the IV&V Software Validation Test through and including the Site Acceptance Test.

7.2.1 Test Hierarchy

[[

]]

7.2.2 Test Activities

[[

]]

7.2.3 Vendor Test Submittal

Test submittals include 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 SOFTWARE VALIDATION TEST – INDEPENDENT VERIFICATION AND VERIFICATION TEAM (IVVT)

[[

]]

7.4 SYSTEM FACTORY ACCEPTANCE TEST

[[

]]

7.5 [[ ]] TEST

[[

]]

7.6 SITE ACCEPTANCE TEST

[[

]]

7.7 TEST DOCUMENTATION

[[

]]

7.7.1 Test Plans

[[

]]

7.7.2 Test Design Specifications

[[

]]

7.7.3 Test Case Specifications

[[

]]

7.7.4 Test Procedure Specifications

[[

]]

7.7.5 Test Item Transmittal Report

[[

]]

7.7.6 Test Log

[[

]]

7.7.7 Test Incident Reports

[[

]]

7.7.8 Test Summary Reports

[[

]]

8. SQAPM MAINTENANCE

The SPE and SQA Managers are responsible for the maintenance of the SQAPM. The 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 3.5.1.2. When improvements or deficiencies are identified, a Corrective Action Request (CAR) should be used to document the condition in accordance with Corrective Action Process [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. 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 Section 3.2, Management Organization.

9. TABLES & FIGURES

Table 9-1a

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 9-1b

Requirements Phase

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

Table 9-1b

Requirements Phase

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

Table 9-1c

Design Phase

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

Table 9-1c

Design Phase

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

Table 9-1c

Design Phase

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

Table 9-1c

Design Phase

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

Table 9-1d

Implementation Phase

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

Table 9-1d

Implementation Phase

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

Table 9-1d

Implementation Phase

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

Table 9-1e

Test Phase

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

Table 9-1e

Test Phase

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

Table 9-1e

Test Phase

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

Table 9-1f

Installation Phase

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

Table 9-1f

Installation Phase

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

Table 9-1f

Installation Phase

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

Table 9-1g

Operations and Maintenance Phase

Operation & Maintenance Tasks	Operation & Maintenance Task Inputs	Operation & Maintenance Task Outputs	Operation & Maintenance Organization	Review Organization	Associated Quality Tasks
[[					
					]]

Table 9-2 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.

Table 9-2

V&V and SSA Tasks Assigned to Each Software Class

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

Table 9-2

V&V and SSA Tasks Assigned to Each Software Class

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

Table 9-2

V&V and SSA Tasks Assigned to Each Software Class

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

Table 9-2

V&V and SSA Tasks Assigned to Each Software Class

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

Table 9-2

V&V and SSA Tasks Assigned to Each Software Class

Software Life Cycle Processes	Planning			Requirements			Design			Implementation			Test			Installation			Operations & Maintenance		
Software Classification	**Q**	**N3**	**N2**	**Q**	**N3**	**N2**	**Q**	**N3**	**N2**	**Q**	**N3**	**N2**	**Q**	**N3**	**N2**	**Q**	**N3**	**N2**	**Q**	**N3**	**N2**

Table 9-2

V&V and SSA Tasks Assigned to Each Software Class

Software Life Cycle Processes	Planning			Requirements			Design			Implementation			Test			Installation			Operations & Maintenance		
Software Classification	Q	N3	N2	Q	N3	N2	Q	N3	N2	Q	N3	N2	Q	N3	N2	Q	N3	N2	Q	N3	N2
																					]]

[[

]]

Table 9-3

Problems and Corrective Action Reporting

Step	Scenario	Identified by	Documentation
[[			

Table 9-3

Problems and Corrective Action Reporting

Step	Scenario	Identified by	Documentation

Table 9-3

Problems and Corrective Action Reporting

Step	Scenario	Dispositioned / Evaluated by	Documentation

Table 9-3

Problems and Corrective Action Reporting

Step	Scenario	Dispositioned / Evaluated by	Documentation

Table 9-3

Problems and Corrective Action Reporting

Step	Scenario	Accepted / Closed by	Documentation

Table 9-3

Problems and Corrective Action Reporting

Step	Scenario	Accepted / Closed by	Documentation
			]]

Table 9-4

Configuration Items

Configuration Items	Format	Retention Medium
Planning Phase		
[[		
		]]
Requirements Phase		
[[		

Table 9-4

Configuration Items

Configuration Items	Format	Retention Medium
		.]]
Design Phase		
[[		
		.]]
Implementation Phase		
[[		

Table 9-4

Configuration Items

Configuration Items	Format	Retention Medium
		.]]
Test Phase		
[[		
		.]]
Installation Phase		

Table 9-4

Configuration Items

Configuration Items	Format	Retention Medium
[[		
		]]
Operation and Maintenance Phase		
[[		
		]]
Retirement Phase		
[[		

Table 9-4

Configuration Items

Configuration Items	Format	Retention Medium
		.]]

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:

CONFIGURATION MANAGEMENT:	
OBJECTIVES:	
SCOPE:	
ITEMS TO BE BASELINED:	APPROVED DATE:

V&V AND SSA SUMMARY:
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:

Extension	Example
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 Subsection 2.2.4								

IEEE Standards from Subsection 2.2.4								

IEEE Standards from Subsection 2.2.4								

IEEE Standards from Subsection 2.2.4								

IEEE Standards from Subsection 2.2.4								

IEEE Standards from Section 2.4								

IEEE Standards from Section 2.4								
								]]

APPENDIX B ACRONYMS AND ABBREVIATIONS

The following acronyms and abbreviations are used throughout this plan.

Acronym	Meaning
ASL	Approved Suppliers List
ASME	American Society of Mechanical Engineers
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
CDA	Critical Digital Asset
CEO	Chief Executive Officer
CFR	Code of Federal Regulations
CI	Configuration Item
CIR	Customer Issue Resolution
CM	Configuration Management
CMM	Configuration Management Manager
CMS	Configuration Management System
COTS	Commercial-Off-The-Shelf
CPU	Central Processing Unit
CySA	Cyber Security Assessment Report
CySPP	Cyber Security Program Plan
CyST	Cyber Security Team

Acronym	Meaning
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
EIA	Electronic Industries Alliance
EMC	Electromagnetic Compatibility
EOP	Engineering Operating Procedure
EPRI	Electrical Power Research Institute
EPROM	Erasable Programmable Read-Only Memory
ERM	Engineering Review Memorandum
ESBWR	Economic Simplified Boiling Water Reactor
FDDR	Field Deviation Disposition Request
FDI	Field Disposition Instruction
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
HSI	Human System Interface
HSS	Hardware/Software Specification
ISCPS	Intra-System Communication Protocol Specification
I&C	Instrumentation and Controls
IEEE	Institute of Electrical and Electronic Engineers
I/O	Input/Output
IP	Implementation Plan

Acronym	Meaning
IR	Inspection Report
ISO	International Standards Organization
IV&V	Independent Verification and Validation
IVVT	Independent Verification and Validation Team
LD	Logic Diagram
LLC	Limited Liability Corporation
LTR	Licensing Topical Report
MCR	Main Control Room
[[	]]
MMI	Man Machine Interface
MMIS	Man Machine Interface System
N/A	Not Applicable
N-DCIS	Nonsafety-related – 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

Acronym	Meaning
PRA	Probabilistic Risk Assessment
PRM	Process radiation Monitor
PWP	Project Work Plan
Q-DCIS	Safety-related - Distributed Control and Information System
QA	Quality Assurance
QCE	Quality Control Engineer
RCCE	Responsible Configuration Control Engineer
RE	Responsible Engineer
RG	Regulatory Guide
RM	Responsible Manager
RMCN	Review Memorandum Change Notice
RSE	Responsible System Engineer
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

Acronym	Meaning
SFAT	System Factory Acceptance Test
SFT	Software Function Test
SFTR	Software Functional Test Report
SIntP	Software Integration Plan
SIP	Software Installation Plan
SITT	System Installation Test Team
SMP	Software Management Plan
SMPM	Software Management Program Manual
SOMP	Software Operations and Maintenance Plan
SPE	Software Project Engineering
SQA	Software Quality Assurance
SQAP	Software Quality Assurance Plan
SQAPM	Software Quality Assurance Program Manual
SRP	Standard Review Plan
SRS	Software Requirements Specification
SSA	Software Safety Analysis
SSP	Software Safety Plan
SST	Software Safety Team
STP	Software Test Plan
STrngP	Software Training Plan
SVT	Software Validation Testing
SVTP	Software Validation Test Plan
SVVP	Software Validation and Verification Plan
SyRS	System Requirement Specification
TBD	To Be Determined
TPE	Technical Project Engineer

Acronym	Meaning
TR	Topical Report
TSL	Training Services Lead
UIS	User Interface Specification
V&V	Verification and Validation
WBS	Work Breakdown Structure

APPENDIX C DEFINITIONS

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

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	In software engineering, computer instructions and data definitions expressed in a programming language or in a form output by assembler, compiler, or other translator [IEEE 610.12].
Code Review	A meeting at which software code is presented to project personnel, managers, users, customers, or other interested parties for comment or approval [IEEE 610.12].
Coding	In software engineering, the process of expressing a computer program in a programming language [IEEE 610.12].
Commitment Tracking System	System used to manage the Conditions Adverse to Quality (CAQs). A Corrective Action Request (CAR) is used to document a CAQ, or an opportunity for process/product improvement, provide for timely evaluation, and record objective evidence of actions taken. Corrective Action Process [2.3(2.v)] specifies the responsibilities for actions to promptly identify, record and correct, as appropriate, CAQs, and to assure that these conditions do not affect the quality of a product or service.
Component	One of the parts that make up a system. A component may be hardware of software and may be subdivided into other components [IEEE 610.12].
Computer Language	A language designed to enable humans to communicate with computers [IEEE 610.12].
Configuration Control	An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification [IEEE 610.12].
Configuration Item	An aggregation of hardware, software, design documents or procedures that is designated for configuration management and treated as a single entity in the configuration management process [IEEE 610.12].

Term	Definition
Critical Digital Asset	A digital device or system that plays a role in the operation or maintenance of a critical system and can impact the proper functioning of a critical system.
Criticality Analysis	The degree of impact that a requirement, module, error, fault, failure, or other item has on the development or operation of a system. A method used to determine the impact of the software product on the system & environment as a whole and thereby determine the software importance (i.e. safety-related, non-safety-related, etc.).
Design Documentation	Design Documentation is information recorded about a specific life cycle activity. Documentation includes software life-cycle design outputs and software life cycle process documentation. A document may be in written or electronic format, and may contain text, illustrations, tables, computer files, program listings, binary images, and other forms of expression. A document for an activity may be packaged with documents for other activities, or documents for non-software life cycle activities. A document for an activity may be divided into several individual entities.
Design Output	Documents, such as drawings and specifications, that define technical requirements of structures, systems, and components. For software, design outputs include the products of the development process that describe the end product that will be installed a nuclear power plant. The design outputs of a software development process include SRS, SDD, hardware and software architecture designs, code listings, system build documents, installation configuration tables, O&M manuals, and training manuals.
Design Phase	The *phase* in the software life cycle during which the designs for architecture, software components, interfaces, and data are created, documented, and verified to satisfy requirements [IEEE 610.12].
Design Record File	A formal controlled information record under GEH procedures for in-progress and completed engineering work which is retained and from which work can be retrieved.
Design Reviews	Formal, design adequacy evaluations which are performed by knowledgeable persons other than those directly responsible and accountable for the design in accordance with 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.

Term	Definition
Deviation	A departure from a specified requirement.
Documentation	A collection of documents on a given subject [IEEE 610.12].
Error	An incorrect step, process, or data definition [IEEE 610.12].
Failure Mode And Effects Analysis	A tabular method of providing traceability from the modes by which a system may fail and the effect of that failure on the ability of the system to perform its function, or the ability of a collection of systems to recover from the failure.
Fault Tree	A pictorial method of providing traceability from the modes by which a system may fail and the effect of that failure on the ability of the system to perform its function, or the ability of a collection of systems to recover from the failure.
Field Deviation Disposition Request	Field Deviation Disposition Request (FDDR) is used for documenting and disposition of the technical position for a deviation required in the field in supplied hardware, software, or services (see Field Deviation Disposition Request [2.3(2.q)]).
Firmware	The combination of a hardware device and computer instructions and data that reside as read-only software on that device [IEEE 610.12].
Functional Testing	A system/software test methodology that is derived from external specifications and requirements of the system. Such testing ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions [IEEE 610.12]. Methods for functional testing include random testing and testing at boundary values. It verifies the end results at the system level, but does not check the implementation techniques, nor does it assume that all statements in the program are executed.
Implementation Phase	The *phase* in the software life cycle during which a software product is created from design documentation and debugged [IEEE 610.12].
Independent Verification And Validation (IV&V)	Verification and Validation performed by an Organization that is technically managerially and financially independent of the development Organization [IEEE 610.12] and RG 1.168 Section C3 [2.2.3].
Installation Phase	The *phase* in the software life cycle during which the software product is installed into its operational environment and tested to ensure that it performs as intended [IEEE 610.12].

Term	Definition
Instrument	A hardware device used for analytical or control functions and usually containing an embedded microprocessor(s).
Integration Testing	Testing in which software elements, hardware elements, or both are combined and tested to evaluate the interaction between them [IEEE 610.12].
Interface	1) A shared boundary across which information is passed. This definition is interpreted broadly to include design interfaces between participating design organizations. 2) A hardware or software component that connects two or more other components for the purpose of passing information from one to the other. 3) To connect two or more components for the purpose of passing information from one to the other. 4) To serve as a connecting or connected component as in (2). [IEEE 610.12 as modified by RG 1.69
Metric	A quantitative measure of the degree to which a system, component, or process possesses a given attribute [IEEE 610.12].
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].
Module Testing	Testing of individual hardware or software units or groups of related units [IEEE 610.12].
Operations and Maintenance Phase	The *phase* in the software life cycle during which the software product is functioning in its operational environment, monitored for satisfactory performance and modified as necessary to correct problems or to respond to changing requirements [IEEE 610.12].
Package	A separately compliable software component consisting of related data types, data objects and sub-programs [IEEE 610.12].
Path Testing	Testing designed to execute all or selected paths through a computer program [IEEE 610.12].
Planning Phase	The initial *phase* of a software development project, in which project scope, purpose, strategy, schedule and milestones are established and user needs through documentation (for example, system definition documentation and procedures) are described and evaluated.

Term	Definition
Plant Process Control and Monitoring Software	Software (including firmware) that controls, monitors, interfaces, or communicates with real time operating digital computer-based plant process control and monitoring devices, equipment and systems located within a nuclear power plant. This includes the software within any other digital equipment of a nuclear power plant, the changes to which after release would constitute a design change. Other digital computer-based systems that are not plant process control and monitoring systems, components, devices and equipment may contain software (including firmware) but are not within the scope of the software plans. These are: • Software (including firmware) within plant security equipment (e.g., perimeter intrusion detection processors, CCTV processors, security access computer and intelligent multiplexers, hand geometry and card reader processors, infra-red detection processors, etc.) subject to the requirements of 10 CFR 73.55. • Communications software (including firmware) such as telephone private and branch exchange switches as well as microprocessor-based public address software. • Software (including firmware) that is not within the scope of the certified design includes but is not limited to; Health Physics radiological monitoring and access control software, Chemistry laboratory equipment and radiological effluents tracking software, Emergency Planning software for dose assessment or other accident response functions, etc.
Procedure	A course of action to be taken to perform a given task [IEEE 610.12].
Process	A sequence of steps performed for a given purpose, e.g., the software development process [IEEE 610.12].
Project Management Plan	A document that describes the technical and management approach to be followed for a project. The plan typically describes the work to be done, the resources required, the methods to be used, the procedures to be followed, the schedules to be met, and the way that the project will be organized [IEEE 610.12].
Regression Testing	Selective re-testing of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements [IEEE 610.12].

Term	Definition
Requirement	A condition or capability that must be met or possessed by a system or system component to satisfy a contract standard specification or other formally imposed documents [IEEE 610.12]. In specifying requirements, the word **shall** is used to indicate mandatory requirements and from which no deviation is permitted **('shall'** and **'required to'** are equivalent in meaning**)**. Requirements are not specified with the word **should.** Instead, it is used to indicate that a recommended course of action and is particularly suitable, without mentioning or excluding other courses of action; Also, a certain course of action is preferred but not necessarily required; Also, that (in the negative form) a certain course of action is not prohibited ('should' and 'recommended' are equivalent in meaning).
Requirements Analysis	The process of studying user needs to arrive at a definition of system, hardware, or software requirements [IEEE 610.12].
Requirements Phase	The *phase* in the software life cycle during which the requirements for a software product are defined and documented [IEEE 610.12].
Requirements Traceability Analysis	The process of tracing the life of a requirement, in both forward and backward direction, using independent verification and traceability matrix to analyze the identified relationships from its source, through design, development, testing and installation to assure the correctness, completeness and accuracy of the software product.
Responsible Configuration Control Engineer	The person assigned responsibility for the configuration management of the I&C software products.
Responsible Engineer	The person responsible for a given technical item, e.g., the design and development of the documentation.
Responsible Technical Project Engineer	The person with overall technical responsibility for ensuring that the hardware and software design of a software product meets the specified requirements.
Responsible Verifier	The Responsible Verifier(s) is an individual who has the independence described in Independent Design Verification [2.3(2.i)] for verifications, or in Deferred Design Verification [2.3(2.j)] for deferred verifications of design process and the accompanying documents.

Term	Definition
Retirement	Permanent removal of a system or component from its operational environment [IEEE 610.12].
Safety-Related	Safety-related structures, systems, components, and parts provide safety-related functions necessary to assure: • The integrity of the reactor coolant pressure boundary; or • The capability to shut down the reactor and maintain it in a safe shutdown condition; or • 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.
Simulation	A model that behaves or operates like a given system when provided a set of controlled inputs [IEEE 610.12].
Software Class N2	Nonsafety-related system software whose failure cannot adversely affect a safety-related function.
Software Class N3	Nonsafety-related systems software whose failure could challenge safety systems as defined below: • 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)]. • Software that is intended to mitigate the result of an accident. • Software that is intended to recover from the result of an accident.
Software Class Q	Software performs functions classified per Safety-Related Classification determination process [2.3(2.t)] as safety-related.
Software Development Process	The process by which user needs are translated into a software product. The process involves translating user needs into software 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].

Term	Definition
Software Life Cycle	The period of time that begins when a software product is conceived and ends when the software is no longer available for use [IEEE 610.12].
Software Life Cycle Phase	The division of the software life cycle into discrete logical units. The I&C software life cycle is divided into eight phases, namely, Planning, Requirements, Design, Implementation, Test, Installation, Operation & Maintenance and Retirement.
Software Module	See Module
Software Package	See Package
Software Unit	See Module
Source Code	Computer instructions and data definitions expressed in a form suitable for input to an assembler, compiler, or other translator.
Statement Testing	Testing designed to execute each statement or a computer program [IEEE 610.12].
Stress Testing	Testing conducted to evaluate a system or component at or beyond the limits of its specified requirements [IEEE 610.12].
Supplemental Document	Controlled documents that are referenced or used in conjunction with this plan. These are the enabling documents that either augment or enable the performance of the activities stated in this plan.
Support Software	Software that aids in the development or maintenance of other software; for example, compilers, loaders, and other utilities [IEEE 610.12].
Supporting Document	Controlled documents used in the production of this plan. These documents form the design basis for the activities stated in this plan.
System Testing	Testing conducted on a complete, integrated system to evaluate the systems compliance with its specified requirements [IEEE 610.12].
Technical Project Engineer	The person with overall technical responsibility for ensuring that the hardware and software design of a software product meets the specified requirements.
Test Case	A set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement [IEEE 610.12].

Term	Definition
Test Item	A software item that is an object of testing [IEEE 610.12].
Test Log	A chronological record of all relevant details about the execution of a test [IEEE 610.12].
Test Objective	An identified set of software features to be measured under specified conditions by comparing actual behavior with the required behavior described in the software documentation [IEEE 610.12].
Test Phase	The *phase* in the software life cycle during which the components of a software product are integrated with the hardware and evaluated to determine whether or not performance requirements have been satisfied [IEEE 610.12].
Test Plan	A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do such task, and any risks requiring contingency planning [IEEE 610.12].
Traceability Matrix	A matrix that records the relationship between two or more product specifications (i.e., design documentation) of the development process (e.g., a matrix that records the relationship between the requirements and the design of a given software component) [IEEE 610.12].
Unit Testing	See Module Testing
User Interface	An interface that enables information to be passed between a human user and hardware or software components of a computer system [IEEE 610.12].
Verification and Validation (V&V)	The process of determining whether the requirements for a system or component are complete and correct, the products of each development phase fulfill the requirements or conditions imposed by the previous phase, and the final system or component complies with specified requirements [IEEE 610.12]

APPENDIX D SOFTWARE CHARACTERISTICS

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

- Software functional characteristics
- Software development process characteristics

Functional Term	Definition
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] 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.

Functional Term	Definition
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 TASK DEFINITIONS

Term	Definition
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	Evaluate database design as part of a design review process could include the following: Physical Limitations Analysis 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. Index vs. Storage Analysis 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. Data Structures Analysis 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. Backup and Disaster Recovery Analysis Review the methods employed for backup against the requirements for data recovery and system disaster recovery and identify deficiencies [IEEE-1012].

Term	Definition
Data Flow Analysis	Evaluate data flow diagrams as part of a design review process. This could include the following: Symbology Consistency Check. 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. Flow Balancing. 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. Confirmation of Derived Data. 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. Keys to Index Comparison. 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].
Design Phase Inspection	Design Phase Baseline Review
Disaster Recovery Plan Assessment	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: Identification of the disaster recovery team and a contact list. Recovery operation procedures. Procedure for establishing an alternative site including voice and data communications, mail, and support equipment. Plans for replacement of computer equipment Establishment of a system backup schedule Procedures for storage and retrieval of software, data, documentation, and vital records off-site. Logistics of moving staff, data, documentation, etc [IEEE-1012].
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].

Term	Definition
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 verity 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].
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.

Term	Definition
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 he plan be coordinated with the acquirer. Incorporate the project software integrity level scheme into the planning process [IEEE-1012].
Previously Developed Software Assessment	Only formal assessments of existing software will be addressed. 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). The assessment can be formative, summative, objective, subjective, criterion-referenced, and/or norm-referenced.
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].
Security Assessment	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]

Term	Definition
Simulation Analysis	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].
Sizing and Timing Analysis	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: CPU load. Random access memory and secondary storage (e.g., disk, tape) utilization. Network speed and capacity. Input and output speed. Sizing and timing analysis is started at software design and iterated through acceptance testing [IEEE-1012].
Software Design Evaluation	Evaluate the design elements (SDD) for correctness, consistency, completeness, accuracy, readability, and testability [IEEE-1012].
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.

Term	Definition
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
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].

Term	Definition
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].

Summary of Changes in SQAPM Rev 5 From SQAPM Rev 4

ITEM	LOCATION	CHANGE
Entire Document		
1.	Entire document	Minor editorial changes for legibility.
Section 1		
2.	Section 1.1, 1st para.	Deleted "I&C system" and replaced with "digital computer-based plant process control and monitoring software" to incorporate changes per RAI 7.1-142.
3.	Section 1.2, Last para.	Added two phrases; "digital computer-based plant process control and monitoring software. This includes" and "It also includes non-DCIS real time plant systems such as but not limited to local fire protection systems, local programmable logic controllers (PLCs), digital standalone controllers and indicators, inverters and battery chargers, electrical distribution digital protective relays, switchgear instrumentation and circuit breaker controllers, meteorological monitoring systems, digital hygrometers, digital salinity cells, digital pH meters, digital conductivity cells, digital dissolved gas monitors, digital area explosive gas monitors, etc." to incorporate changes per RAI 7.1-142.
4.	Section 1.5, 2nd to last para.	Deleted "I&C system" and replaced with "digital computer-based plant process control and monitoring software" to incorporate changes per RAI 7.1-142.
5.	Section 1.5, Fig. 1	Corrected reference in 1st diamond of figure 1 and added cross-reference to Note 1 for consistency's sake.
Section 2		

ITEM	LOCATION	CHANGE
15.	Sub-section 5.3.1.1, 1st sentence, 1st para.	Deleted the word "it" for grammatical correctness.
16.	Sub-section 5.3.1.6, Table 5.3.1.6-1	In the activity for task 3, specified that system requirements review is performed of System Design Specification for consistency with Table 5.6-1 in SMPM [2.3(1.a)].
17.	Sub-section 5.3.1.6, Table 5.3.1.6-1	Added a note to specify who performs V&V activities for N3 and N2 acquired software consistent with section 3.9.
18.	Sub-section 5.3.2.1, Table 5.3.2.1-1	In last bullet of the outputs, changes sentence for grammatical correctness.
19.	Sub-section 5.3.4.3, 1st para.	Deleted reference to component testing when SVT and SFAT are being performed and changed to system testing for consistency with sections 7.3 and 7.4. Component level testing is performed to the SFT in accordance with SMPM [2.3(1.a)], sub-section 6.11.2.1.
20.	Sub-section 5.3.5.4, 1st sentence	Corrected criticality cross-reference from sub-section 4.3.3.1 to 4.3.4.1.
21.	Sub-section 5.3.7.6, 1st sentence Table 5.3.7.6-1, Task 1	Added phraseology to indicate task includes both the generation and the evaluation of the Installation Configuration Table for the sake of consistency with the title of this sub-section. Originally the verbiage indicated just evaluation, NOT generation.
22.	Sub-section 5.3.7.8, 1st sentence Table 5.3.7.8-1, 1st Task	Revised verbiage to include both hazard and risk analysis consistent with title of sub-section.
23.	Sub-section 5.3.8.8, 1st sentence 2nd and 3rd sentences, and Table 5.3.8.8-1	Corrected cross-reference in 1st sentence from sub-section 4.3.6 to sub-section 4.3.7. Revised verbiage to include both hazard and risk analysis consistent with title of sub-section.
24.	Sub-section 5.3.8.9, 1st sentence	Changed the verb "identify" to the past participle "identified" for grammatical correctness.
25.	Sub-section 5.4.3	Corrected cross-reference to test report from sub-section 7.7.3 to sub-section 7.7.8.

ITEM	LOCATION	CHANGE
Section 8		
33.	Section 8, 1st para., 2nd sentence	Corrected cross-reference from subsection 3.5.2 to 3.5.1.2.
Section 9		
34.	Tables 9-1a through g, last row	The system design task associated with the quality task "Planning Baseline Review", "Requirements Baseline Review", etc., was not specified; this is now corrected in Tables 9-1a through g be "Baseline Review".
35.	Table 9-1b, 5th row	Review organization cell for the system design task of "Develop SyRS RTA" was NOT specified for various classes of software; this is now corrected.
36.	Table 9-1c, 6th row	Development organization for the SVT development task was specified to be SPE instead of the IVVT within SPE; this is now corrected.
Section 10		
37.	None	None
Appendices		
38.	Appendix C, Terms	Added definition for "Plant Process Control and Monitoring Software" to incorporate changes per RAI 7.1-142.