GE Energy
Nuclear

NEDO-33228
Class I
eDRF# 0000-0046-8503

January 2006

LICENSING TOPICAL REPORT

ESBWR I&C
SOFTWARE VERIFICATION AND VALIDATION PLAN

Copyright 2006 General Electric Company

INFORMATION NOTICE

This document NEDO-33228 contains no proprietary information.

IMPORTANT NOTICE REGARDING CONTENTS OF THIS REPORT PLEASE READ CAREFULLY

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

Table of Contents

1 **Purpose** 8
- 1.1 Scope 8
- 1.2 Application 9

2 **Referenced Documents** 9
- 2.1 Supporting Documents 9
- 2.2 Supplemental Documents 10
- 2.3 Codes and Standards 11
  - 2.3.1 Electrical Power Research Institute (EPRI) 11
  - 2.3.2 Institute of Electrical and Electronic Engineers (IEEE) 11
  - 2.3.3 U.S. Nuclear Regulatory Commission (NRC) 11
  - 2.3.4 International Standards 11

3 **Acronyms and Definitions** 12
- 3.1 Acronyms 12
- 3.2 Definitions 14
  - 3.2.1 Software Life Cycle Phases 14
  - 3.2.2 General 15

4 **Verification and Validation Overview** 17
- 4.1 Organization 17
- 4.2 Organizational Interfaces 18
- 4.3 Master Schedule 18
- 4.4 Personnel Resources Summary 19
- 4.5 Roles and Responsibilities 19
  - 4.5.1 Qualification of Responsible Individuals 19
- 4.6 Techniques, Methodologies and Tools 21
  - 4.6.1 Verification and Validation 21
  - 4.6.2 Use of Metrics 22
  - 4.6.3 Tools Used to Aid V&V Activities 22
- 4.7 Assessment Strategy 22
- 4.8 SVVP Cross Reference to Standards 22

5 **Management of SVVP Tasks** 23
- 5.1 Management Review 23
- 5.2 Review Support 23

5.3 Risks and Assumptions 24

6 Life Cycle Verification and Validation Tasks 24

6.1 General V&V Task Requirements 24

6.1.1 Inputs 24
6.1.2 Method 24
6.1.3 Acceptance Criteria 24

6.2 Planning Phase 25

6.2.1 Software Development Plans 25
6.2.2 System Definition Documentation 25

6.3 Requirements Phase 26

6.3.1 Requirements Documentation 26
6.3.2 Software Validation Test Plan 26
6.3.3 V&V Requirements for Acquired Software 27
6.3.4 Requirements Phase Safety Analysis 30
6.3.5 Requirements Traceability Analysis 30
6.3.6 Software Requirements 31
6.3.7 Software Requirements Interface Analysis 31
6.3.8 System Test Plan 31
6.3.9 Acceptance Test Plan 31
6.3.10 Software Architecture Description 32

6.4 Design Phase 32

6.4.1 Design Safety Analysis 32
6.4.2 Design Traceability Analysis 32
6.4.3 Design 33
6.4.4 Design Interface Analysis 33
6.4.5 Component Test Plan 33
6.4.6 Integration Test Plan 33
6.4.7 Test Design 34
6.4.8 User's Manual & Documentation 34
6.4.9 Software Code Style Guide 35
6.4.10 Validation Test Procedures 35
6.4.11 Design Metrics 35

6.5 Implementation Phase 35

6.5.1 Implementation Safety Analysis 35
6.5.2 Source Code Traceability Analysis 36
6.5.3 Source Code 36
6.5.4 Source Code Interface Analysis 36
6.5.5 Source Code Documentation 37
6.5.6 Module Testing 37
6.5.7 Test Procedure (Module Testing) 37
6.5.8 Module Test Report 37
6.5.9 Implementation Phase Metrics 38

6.6 Integration Phase 38

6.6.1 Test Safety Analysis 38
6.6.2 Integration Testing 39
6.6.3 System Build Documentation 40
6.6.4 Integration Test Metrics 40
6.7 Validation Test Phase 40
6.7.1 Validation Safety Analysis 41
6.7.2 Validation Test Plan 41
6.7.3 Validation Test Report 41
6.7.4 Validation Test Metrics 42
6.8 Installation Phase 42
6.9 Operations and Maintenance Phase 42
6.10 Baseline Reviews 42
6.10.1 Planning Phase 42
6.10.2 Requirements Phase 43
6.10.3 Design Phase 43
6.10.4 Implementation Phase 43
6.10.5 Integration Phase 44
6.10.6 Validation Test Phase 44

7 Software Verification and Validation Reporting 45
7.1 V&V Reporting 45
7.2 Anomaly Reporting 45
7.3 Engineering Management Report 45
7.4 Final V&V Final Report 45

8 Verification and Validation Administrative Procedures 46
8.1 VVT Schedule 46
8.2 Nonconformance & Anomaly Reporting and Resolution 46
8.3 Task Iteration Policy 46
8.4 Deviation Policy 47
8.5 Control Procedures 47
8.6 Standards, Practices, and Conventions 47

List of Tables

TABLE 1 LIFE CYCLE PHASE CROSS-REFERENCE....48
TABLE 2 SVVP CROSS-REFERENCE TO SRP SECTION 7, BTP HICB-14 [2.3.3(3)]....49
TABLE 3 SVVP CROSS REFERENCE TO IEEE 1012 [2.3.2(1)]....51
TABLE 4 LIFE CYCLE PHASE OUTPUT ASSESSMENT REQUIREMENTS....68

List of Figures

FIGURE 1 DESIGN ORGANIZATIONS AND SPE INTERFACE CHART.....73

1 Purpose

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

1. the design documents meet the requirements outlined in the Man-Machine Interface System and HFE Design Implementation Plan [2.1(2)], the SMP [2.2(1)], the SCMP [2.2(2)] and SSP [2.2(4)].

2. the developed software meets its specified requirements,

3. the developed software performs its intended functions correctly,

4. the developed software performs no unintended functions, and

5. the quality and reliability standards for the software-based product are achieved.

Software verification and validation (V&V) activities shall be performed as part of the ongoing software development process to facilitate the timely detection of errors.

For the DCIS software-based products, software V&V activities shall also be included to analyze and test the software with respect to its hardware interfaces and user interactions.

This plan is designed to remain in effect throughout the entire life cycle of the DCIS software-based product. Exceptions from this plan may be accepted if the product quality level is maintained. Any deviations from this plan shall be approved by the Responsible Technical Project Engineer (RTPE), the Responsible Software Safety Engineer (RSSE) and Software Project Engineering (SPE). The justification shall be documented in the software project Design Record File (DRF) {see GEEN EOP 42-10.00 [2.2(7c)]}.

1.1 Scope

This plan (SVVP) outlines the formal set of standards and procedures necessary to comprehensively verify and validate Quality Class Q and Class N DCIS software-based products during all phases of the software development life cycle {see SMP [2.2(1)]}.

The V&V activities performed by Software Project Engineering (SPE) are to ensure the quality of the design and development of software-based products and their associated documentation. This type of verification is performed on each design document produced as part of the design and development process in accordance with EOPs 40-7.00 [2.2(7a)] and/or 42-6.00 [2.2(7b)].

The software life cycle phases in the SVVP correspond with those defined in the Software Management Plan [2.2(1)] and are described in Section 3.2.1, "Software Life Cycle Phases." The scope of V&V includes the following:

1. Planning Phase: Plans, System concept, human-hardware-software allocation.

2. Requirements Phase: System, software, computer platform.
3. Design Phase: Software, architecture.
4. Implementation Phase: Software.
5. Integration Phase: System, software.
6. Validation Phase: System, software.

Where:

- "System" refers to characteristics observable at system boundaries
- "Computer Platform" refers to characteristics observable by software
- "Architecture" refers to integrated hardware and software

The V&V tasks, which cover this scope, are described in Section 6, "Life Cycle Verification and Validation Tasks."

The installation activities are outside the scope of this document.

1.2 Application

V&V activity is limited to software prepared by GE and GE vendors for the Project DCIS.

The SVVP is intended to be used by the design organization to prepare life-cycle phase outputs and by SPE in performing Independent V&V of DCIS software life-cycle phase outputs.

2 Referenced Documents

Supporting and supplemental reference documents as well as referenced standards and codes applicable to this V&V plan are outlined in the following sections.

2.1 Supporting Documents

The following supporting documents were used as the controlling documents in the production of this plan. These documents form the design basis traceability for the requirements outlined in this plan.

Document Title	Document Number
1. Project Design Manual (PDM)	
2. Man-Machine Interface System and HFE Design Implementation Plan (MMIS/HFE IP)	NEDO-33217

Document Title	Document Number
3. NP 2010 COL Demonstration Project Quality Assurance Program	NEDO-33181

2.2 Supplemental Documents

Supplemental documents are those documents that are used in conjunction with this document.

Document Title	Document Number
1. Software Management Plan (SMP)	NEDO-33226
2. Software Configuration Management Plan (SCMP)	NEDO-33227
3. Software Development Plan (SDP)	NEDO-33229
4. Software Safety Plan (SSP)	NEDO-33230
5. Software Quality Assurance Plan (SQAP)	NEDO-33245
6. Software Test Plan (STP)	
7. GE Nuclear Energy Engineering Operation Procedures	NEDE-21109
a. 40-7.00 Design Review	
b. 42-6.00 Independent Design Verification	
c. 42-10.00 Design Record File	
d. 40-3.00 Engineering Computer Programs	
e. 42-6.10 Differed Design Verification	
f. 65-2.10 Safety Related Classifications	
8. Software Integration Plan (SIntP)	NEDO-33246
9. Software Installation Plan (SIP)	NEDO-33247
10. Software Maintenance Plan (SMtP)	NEDO-33248
11. Software Training Plan (STgP)	NEDO-33249
12. Software Operations Plan (SOP)	NEDO-33250

2.3 Codes and Standards

The following codes and standards are applicable to the Software Verification and Validation Plan (this plan).

2.3.1 Electrical Power Research Institute (EPRI)

1. EPRI TR-106439, Guidelines on Evaluation and Acceptance of Commercial Grade Digital Equipment in Nuclear Safety Applications

2.3.2 Institute of Electrical and Electronic Engineers (IEEE)

1. IEEE 1012, Standard for Verification and Validation Plans
2. IEEE 1028, Standard for Software Reviews and Audits
3. IEEE 1228, Standard for Software Safety Plans
4. IEEE 7-4.3.2, Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations
5. IEEE 829, Standard for Software Test Documentation
6. IEEE 1008, Standard for Software Unit Testing

2.3.3 U.S. Nuclear Regulatory Commission (NRC)

1. Regulatory Guide (Reg Guide) 1.168, Verification, Validation, Review, and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants
2. NUREG/CR-6101, Software Reliability and Safety in Nuclear Reactor Protection Systems
3. Standard Review Plan (SRP), Section 7, Branch Technical Position (BTP) HICB-14, Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems
4. NUREG/CR-6463, Coding Conventions
5. 10 CFR 50, Appendix B
6. Reg Guide 1.170, Software Test Documentation for Visual Computer Software
7. Reg Guide 1.171, Software Unit Testing
8. CR-6421, A Proposed Acceptance Process for Commercial Off-the-Shelf (COTS) Software in Reactor Applications

2.3.4 International Standards

1. ISO-9000-3, Quality management and quality assurance standards - Guidelines for the application of ISO 9001 to the development, supply and maintenance of software

3 Acronyms and Definitions

3.1 Acronyms

ACE	Abnormal Conditions and Events
BR	Baseline Review
BRT	Baseline Review Team
BTP	Branch Technical Position
COTS	Commercial-Off-The-Shelf
DCIS	Distributed Control and Information System
DRF	Design Record File
ECN	Engineering Change Notice
EOP	Engineering Operating Procedures
ERM	Engineering Review Memorandum
ESBWR	Economically Simplified Boiling Water Reactor
GE	General Electric Company
GEEN	General Electric Energy Nuclear
HFE	Human Factors Engineering
HICB	Instrumentation and Control Branch
HSS	Hardware/Software Specification
ITAAC	Inspection, Tests, Analysis and Acceptance Criteria
MMIS	Man-Machine Interface System
RE	Responsible Engineer
RMMM	Risk Mitigation, Monitoring and Management
SAD	Software Architecture Description
SBD	System Block Diagram
SCMP	Software Configuration Management Plan
SDP	Software Development Plan
SDS	Software Design Specification

SIntP	Software Integration Plan
SIP	Software Installation Plan
SMP	Software Management Plan
SMtP	Software Maintenance Plan
SOP	Software Operations Plan
SPE	Software Project Engineering
SQAP	Software Quality Plan
SRP	Safety Review Plan
SRS	Software Requirements Specification
SSP	Software Safety Plan
STgP	Software Training Plan
STP	Software Test Plan
SVVP	Software Verification and Validation Plan
V&V	Verification and Validation
VVT	Verification and Validation Team

3.2 Definitions

3.2.1 Software Life Cycle Phases

The software life cycle phases defined below correspond to the DCIS Software Management Plan [2.2(1)]. Table 1 is a cross-reference of the life cycle phases defined in the SVVP with those in the Software Management Plan (SMP) [2.2(1)] and standards BTP HICB-14 [2.3.3(3), Section 2.2], IEEE-1012 [2.3.2(1), Section 3], and NUREG/CR-6101 [2.3.3(2), Figure 3-5].

Design Phase	The period of time in the software life cycle during which the designs for software architecture, detailed representation of software components, interfaces, and data are created, documented, and verified to satisfy requirements.
Implementation Phase	The period of time in the software life cycle during which a software product is created from design documentation and debugged.
Installation Phase	The period of time in the software life cycle during which the software product is installed into its operational environment and tested to ensure that it performs as required.
Integration Phase	The period of time 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.
Operations and Maintenance Phase	The period of time in the software life cycle during which the software product is functioning in its operational environment.
Planning Phase	The initial phase of a software development project, in which project scope, purpose, strategy, schedule and milestones are established as well as describing and evaluating user needs through documentation (for example, system definition documentation and procedures).

Requirements Phase — The period of time in the software life cycle during which the requirements, such as detailed functional and performance capabilities, design constraints, and validation criteria are defined and documented.

Validation Phase — The period of time in the software life cycle during which the software product is integrated into its operational environment and tested with approved test plans and procedures to ensure that it performs as required.

3.2.2 General

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.

Baseline — A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.

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 Technical Project Engineer. 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 [2.2(1)], SCMP [2.2(2)]. All Baseline Reviews are performed and documented in accordance with the Software Configuration Management Plan [2.2(2)], the Software Quality Assurance Plan [2.2(5)] and the Software Verification and Validation Plan (this plan).

Component Testing — Testing conducted to verify the implementation of the design for one software element (for example, unit, module) or a collection of software elements.

Design Record File — A formal controlled information record under the GEEN procedures for in-progress and completed engineering work which is retained and from which work can be retrieved.

Digital System	For this SVVP, a digital system is an integrated system of digital hardware and associated Software (embedded or otherwise) that together, comprises a complete system or sub-system.
Documentation	The formal output generated by a task for a particular case (e.g., design output).
Embedded System	A specialized computer system that is part of a larger system or machine. Typically, an embedded system is housed on a single microprocessor board with the programs stored in ROM (Firmware).
Firmware	Software not running on the main processor of a computer will be considered as firmware, i.e., as software embedded in a black box device that can be used and assessed only through the device in which it is embedded.
Integration Testing	An orderly progression of testing in which software elements, hardware elements, or both are combined and tested until the entire system has been integrated.
Responsible Verifier	The person responsible for design documents verification. A responsible verifier is a qualified (i.e., by knowledge and experience) individual who did not participate in the design.
Classifications of Safety-Related (Q) and Nonsafety-Related (N) Hardware and Software	Reference EOP 65-2.10, the Software that performs a nonsafety-related function and its common mode software failure does not defeat a safety-related function, the software is classified as Nonsafety-Related (N), (IEEE 1012 integrity level 1 and 2). The software that performs a safety-related function or its common mode software will defeat a safety-related function, it is classified as Safety-Related (Q), (IEEE 1012 integrity level 3). See Appendix A of SQAP for breakdown of software systems, which are Q class.
System Testing	The process of testing an integrated hardware and software system to verify that the system meets its specified requirements.
Validation	The testing process that ensures that software-based products meet their intended uses and are compliant with system functional, performance and interface requirements.
Verification	Activities performed by knowledgeable personnel independent of those responsible for the software design process to ensure that

design and process outputs meet specified requirements. Design and process outputs may include software development plans, software specifications and manuals, safety analyses. Verification activities include audit, inspection, independent verification, testing, and review of operating experience.

Verification and Validation The verification and validation activities performed by software project engineering (SPE) in accordance with the design process (this SVVP) to ensure the quality of the associated documents produced.

4 Verification and Validation Overview

4.1 Organization

Software Project Engineering (SPE) performs V&V activities. The SPE is described in the Software Quality Assurance Plan (SQAP) [2.2(5)].

SPE is an organizationally independent team responsible for the independent V&V of the design and development and associated supporting documentation of the software-based products and baseline review for the software development. SPE is independent and different from the design team to ensure that the V&V process is not compromised by schedule, budget, or other demands placed on the design organization. The SPE has the following functions with respect to V&V activities:

1. performing the Independent V&V,
2. performing the Baseline Reviews,
3. performing software safety analysis.

The Verification and Validation Team (VVT) is responsible for performing and managing V&V activities on the design and development and associated supporting documentation of the software-based product's design process to:

1. ensure that the design meets the specified requirements,
2. confirm the quality, safety, reliability, and performance of the design, and
3. ensure that the products meet its intended use.

Baseline Review is performed to assure the completeness of each software life cycle phase. The BRT is responsible for performing the baseline review in accordance with the procedures outlined in the Software Configuration Management Plan [2.2(2)] and approve the progression of the design and development activities.

The software safety team performs the software safety analysis in accordance with the software safety plan (SSP) [2.2(4)]. As part of the analysis, the responsible safety engineer shall have part of the scope of the verification of the engineering output and as such shall coordinate this activity with the verification team. The SSP provides the detailed activities of the SSP.

4.2 Organizational Interfaces

The organizational interfaces are described in the Software Quality Assurance Plan (SQAP) [2.2(5)]. Also See Figure 1.

4.3 Master Schedule

The V&V Schedule and contingency planning to identify risk are defined in the Software Development Plan [2.2(3)]. The resource type necessary to support the V&V activities will be defined in the Software Test Plan (STP) [2.2(6)].

The software engineering process is divided into eight (8) discrete life cycle phases {see SMP [2.2(1)]}. The following software life cycle phases have been designated by the SMP for the development of all software-based products:

1. Planning,
2. Requirement Specification,
3. Design,
4. Implementation,
5. Integration,
6. Validation,
7. Installation, and
8. Operation & Maintenance.

With the exception of the Installation and the Operation & Maintenance Phase, a baseline review shall be conducted at the completion of each software life cycle phase {see SCMP [2.2(1)]}. These baseline reviews are designated as major project milestones. During the Maintenance Phase, modification to the design documentation shall be verified and baseline review shall be re-performed for the affected software life cycle phase(s).

All anomalies or nonconformance identified during the V&V, and/or Baseline Reviews shall be resolved by the RE before the software design and development effort can proceed to the next software life cycle phase {unless such progression is approved by the Baseline Review Team, see SCMP [2.2(2)]}.

4.4 Personnel Resources Summary

The Software Quality Assurance Plan (SQAP) [2.2(5)] provides the SPE personnel resource summary.

4.5 Roles and Responsibilities

The roles and responsibilities of SPE members are described in the Software Quality Assurance Plan (SQAP) [2.2(5)].

The Responsible Technical Project Engineer (RTPE) shall:

1. determine the timing of the required V&V and coordinate with the VVT task lead,
2. assure that the V&V are conducted per GEEN EOP 40-7.00, Design Review [2.2(7a)], or EOP 42-6.00, Independent Design Verification [2.2(7b)] ,
3. determine and document the V&V scope and methods, and
4. select the design review team or a verifier in accordance with the EOPs 40-7.00 or 42-6.00.

The RTPE along with the VVT task lead shall periodically review the internal V&V activities to assess the V&V effectiveness, including the following:

1. V&V efforts,
2. Technical accomplishments,
3. Resource utilization, and
4. Timeliness.

The RTPE of a Quality Class Q software-based product and RSSE shall consult with the VVT task lead to ensure the timeliness of V&V activities.

4.5.1 Qualification of Responsible Individuals

Generic qualifications of SPE personnel are described in the SQAP (Sec 2.3) [2.2(5)]. The following sections give qualifications related to specific V&V tasks.

4.5.1.1 Design Reviews

Design reviews are formal, design adequacy evaluations, which are performed by knowledgeable persons other than those directly responsible and accountable for the design. For example, if the design documentation contained Human-System Interface (HSI) specifications, the review team shall include an engineer who is qualified and knowledgeable in Human Factors Engineering (HFE).

4.5.1.2 Independent Design Verifications

Independent Design Verifications are performed to confirm that the design meets specified requirements. The Responsible Verifier(s) shall be qualified by knowledge and experience to perform the verification but did not perform the design. If the design documentation contains Human-System Interface (HSI) specifications, at a minimum at least one verifier shall be knowledgeable in Human Factors Engineering (HFE).

4.5.1.3 Code Reviews

Code reviews are performed to verify that the final software correctly implements the design specified in the SDS, and shall be performed in accordance with the procedure outlined in the SMP {see SMP [2.2(1)]} and the SQAP {see SQAP [2.2(5)]}.

4.5.1.4 Module Testing

Module testing is performed to ensure that the software module executes as specified in the SDS. Module testing shall be performed in accordance with the procedure outlined in the SMP [2.2(1)], SQAP [2.2(5)], and Software Test Plan [2.2(6)]. The software module designer or any qualified software engineer may perform module Testing.

4.5.1.5 Integration Testing

Integration Testing is performed to confirm that:

1. the integrated software modules perform as intended,
2. software, when installed in the prototype or target hardware performs as intended, and
3. proper operation of the interfacing entities as specified in the HSS and SRS for the software-based systems, or in the HSS, SBD and IPS for microprocessor-based systems.

Such testing shall be performed in accordance with the procedures outlined in the SMP [2.2(1)], SQAP [2.2(5)], and STP [2.2(6)]. The software module designer or any qualified software engineer may perform integration Testing. If the system contains user interfaces, at a minimum, the integration testing shall include an engineer who is knowledgeable in Human Factors Engineering (HFE).

4.5.1.6 Validation Testing

Validation Testing is performed to verify that the software-based product is operational and conforms to all functional and performance requirements specified in the HSS. Validation testing shall be performed in accordance with the procedures outlined in Validation Test Procedures and Test Cases Specification {see SMP [2.2(1)]}, SQAP [2.2(5)] and STP [2.2(6)]. Validation Testing shall be conducted by SPE personnel qualified in the technologies and methods used in the design development. At a minimum, Validation Testing shall include an engineer who is knowledgeable in Human Factors Engineering (HFE) to assure that the HSI functions implemented are consistent with the final version of the User's Manual.

4.6 Techniques, Methodologies and Tools

Techniques and methods used to perform Verification and Validation specifically by the SPE are described in the SQAP [2.2(5)]. Additional methods are defined in the following sections. Tools, software, hardware or otherwise, that are used to assist in V&V techniques or methods are generically described as well. The software engineering process shall utilize a combination of the following methods (defined in this plan and the SQAP) to assure the quality of software-based products:

1. V&V, such as, informal reviews, design reviews, independent design verifications and testing.

2. Baseline Reviews.

SST techniques, methods and tools are defined in the SSP [2.2(4)]. See Section 4.1 for SST overall responsibilities for verification activities.

4.6.1 Verification and Validation

V&V are product assurance activities performed to:

1. ensure that the design meets its specified requirements,

2. assure the quality, safety, reliability, and performance of a design, and

3. confirm adherence to the input requirements as specified in the software planning documents.

Such V&V activities substantiate a design and provide controlled, documented confirmation that the design meets its requirements.

V&V shall be conducted by qualified personnel in accordance with the guidelines and procedures outlined in EOP 40-7.00, Design Reviews [2.2(7a)] and EOP 42-6.00, Independent Design Verification [2.2(7b)] and under the management of the SQAP [2.2(5)].

All nonconformances identified during the design reviews shall be documented in the Design Review Report. The RE shall be responsible for resolving these nonconformances. Design Reviews are not considered to be complete until the Design Review Report and all committed action items have been completed and documented. Design Reviews shall be closed with a formal Design Review Closure Letter. A sample of a Design Review Closure Letter and a Design Review Report form are given in Appendices C and D of EOP 40-7.00 [2.2(7a)]. The Design Review Closure Letters and Design Review Reports shall be filed in the software project DRF.

A sample of a Design Verification Guidelines document and a Design Verification Package are given in Appendices A and B of EOP 42-6.00 [2.2(7b)]. The completed Design Verification Records shall be filed in the software project DRF.

4.6.2 Use of Metrics

Use, purpose, technique and reporting of metrics are described in the SQAP [2.2(5)].

4.6.3 Tools Used to Aid V&V Activities

Section 6, Life Cycle Verification and Validation (V&V) Tasks, outlines V&V activities for each phase of the software life-cycle model. Software Verification and Validation Reporting, outlines the expected content of the reports generated by Independent V&V activity.

Test tools used to validate the software against its requirements shall be identified by name along with a description, identification number, qualification status, and version number of tool. The purpose for the tool's use shall also be described. This information shall be recorded for each V&V activity in the test procedures and test report of the specific test. Additional information regarding test tools will be defined in the Software Test Plan [2.2(6)].

4.7 Assessment Strategy

The VVT shall perform an assessment of the software development process (including planning, outputs, implementation, corrective action and rejection) for each software product as described by the SQAP.

Commercial grade software, not developed under 10 CFR 50 Appendix B [2.3.3(5)] [Appendix B] QA program, is considered as COTS. Existing software (developed under an Appendix B QA program for Quality Class Q) is considered, by this plan, as previously developed software. New software being specifically developed for the project is considered to be in-house (if being developed by GE) or 3rd party/vendor software if it is being out-sourced. Systems embedded to the DCIS that are determined to be running firmware and provide a digital interface (either directly to a user or to another system) are considered embedded software systems.

The assessment shall confirm that the process, including process documentation and design output documentation, provides a high level of confidence that the software fulfills plant functional and safety requirements. The assessment shall be based on compliance of the development process commensurate to the quality class of the software package.

The assessment shall be documented in formal reports in accordance with Sections 7.1, V&V Reporting, and 7.2, Anomaly Reporting. The assessment is successfully completed when all identified anomalies are satisfactorily resolved. V&V process metrics (Section 4.6.2, Use of Metrics) shall be used to measure the effectiveness of the design effort in reducing measured defects to an acceptable level.

4.8 SVVP Cross Reference to Standards

Table 2 is a cross-reference of the SVVP with SRP Section 7, BTP HICB-14 [2.3.3(3)]. BTP-14 gives guidance in generic terms and the lack of a direct correlation between document names does not equate to a lack of documents. Since the SMP [2.2(1)] may use a number of different

types of documents to express requirements, BTP 14 shall be considered as guidance for all of these types of requirement documents as applicable.

Table 3 is a cross-reference of the SVVP with IEEE 1012 [2.3.2(1)]. IEEE 1012 expresses requirements in generic terms and the lack of a direct correlation between document names does not equate to a lack of documents. Since the SMP may use a number of different types of documents to express requirements, IEEE 1012 should be considered as guidance for all of these types of requirement documents as applicable.

5 Management of SVVP Tasks

The management of V&V tasks and activities are the responsibility of the VVT task lead as described in Section 4.5. The SPE Task lead has the overall responsibility for VVT activities and shall periodically review VVT activity to ensure continued effectiveness and support of the Independent V&V effort.

5.1 Management Review

The SPE task lead shall review V&V Review Reports, V&V schedules, and consult with the VVT task lead to assess VVT effectiveness. The reviews shall be conducted at least once every four months, and shall include assessments of:

1. V&V efforts
2. Technical accomplishments
3. Resource utilization
4. Future planning
5. Risk management
6. Review of open issues.

The SPE task lead shall recommend changes to V&V tasks as required to improve the V&V effectiveness.

The SPE task lead shall document all management review activities and submit the records to the project DRF.

5.2 Review Support

The SPE task lead shall assure that qualified V&V Team Members are available for the timely execution of SVVP tasks.

The VVT Task lead shall assign V&V Team Members for specific review activities.

V&V Team Members may use qualified individuals from external or 3rd party organizations to meet varying needs of schedule and/or expertise. In all cases V&V Team Members shall be knowledgeable in the technologies and methods used in the design development and qualified by education and experience to perform the V&V task.

The use of external or 3rd party organization personnel for V&V activity is restricted as follows:

1. The individual's Responsible Manager shall agree to the assignment of an individual to the VVT for the performance of the specific task. The Responsible Manager may not place time constraints on task completion.

2. The Responsible Manager shall provide the VVT with a statement that the individual is qualified to perform the specific task.

3. The individual shall be provided training in the requirements of the V&V activity.

5.3 Risks and Assumptions

Generic risks to the implementation of V&V activities are outlined in the SQAP [2.2(5), Sec 2.5.2].

6 Life Cycle Verification and Validation Tasks

Section 6.1 describes the general requirements for Independent V&V activities. Sections 6.2 to 6.9 provide details for the activities in each software life cycle phase.

6.1 General V&V Task Requirements

V&V shall be conducted on all configuration items developed throughout the software life cycle phases as described in this plan.

For Quality Class Q software V&V, a software cross reference (traceability) matrix shall be used to show the linkage between design requirements outlined in the input requirements documents and the implementation of these requirements in the design implementation documents.

The Independent V&V inputs, acceptance criteria and outputs for each task of the overall Life Cycle V&V are described in Sections 6.1.1 through 6.1.3.

6.1.1 Inputs

Table 4 lists the inputs required for each V&V activity.

6.1.2 Method

Table 4 specifies the evaluation method for each V&V activity.

6.1.3 Acceptance Criteria

Table 4 lists the references for the acceptance criteria for each V&V activity.

6.2 Planning Phase

The following sections address the required V&V tasks unique to the planning phase for the project organization. Input and output specifics can be located in Table 4. Generic or phase independent items concerning the planning phase are located, along with the other phase independent items, in Sections 4, 5, 7 and 8 as needed.

6.2.1 Software Development Plans

V&V shall be performed on the planning phase design outputs to ensure that:

1. a software quality assurance process has been defined for the project, and this process meets the required codes and standards,
2. project schedule and timely milestones have been established.

The VVT shall perform a verification of software development planning documents to confirm compliance with applicable standards. The software development plans include:

1. MMIS Interface System and HFE Implementation Plan (MMIS/HFE IP)
2. Software QA Plan (SQAP)
3. Software Management Plan (SMP)
4. Software Verification and Validation Plan (SVVP)
5. Software Configuration Management Plan (SCMP)
6. Software Safety Plan (SSP)
7. Software Development Plan (SDP)
8. Software Integration Plan (SIntP)
9. Software Installation Plan (SIP)
10. Software Maintenance Plan (SMtP)
11. Software Test Plan (STP)
12. Software Operations Plan (SOP)

6.2.2 System Definition Documentation

Verification of System Definition Documentation, as listed in the SMP, is outside the review scope of this plan. Such documents are developed in accordance with 10 CFR 50, Appendix B, QA program and are not considered as software documents themselves, rather they are

considered as inputs to software documents and so applicable standard requirements do not need to be considered.

6.3 Requirements Phase

V&V activity for the requirements phase of the project shall include tasks as described in the following sections. The described task should be considered as a minimum and the VVT shall have the authority to expand these tasks at its discretion.

6.3.1 Requirements Documentation

The VVT shall verify the design organization's software requirements documentation of commercial grade digital equipment for nuclear applications to confirm the following items have been considered:

1. Hardware/Software Specification
2. System Block Diagram
3. Software Requirements Specification
4. Instrument Performance Specification
5. Sub-System Schematic
6. External Data Communication Specification
7. Hardware Software Interface (HSI) Design Specifications (from HFE).

6.3.2 Software Validation Test Plan

The Validation Team shall write and the Verification Team shall verify the software validation test plan in accordance with the STP to be used in validation of digital equipment for nuclear applications to confirm:

1. Testability, readability, consistency, completeness and accuracy.
2. Traceability (forward and backwards) of test criteria for (Quality Class Q software) to software requirements (function, informational and behavioral).
3. Adherence to Table 4.

The validation test plan may be completed at the requirements phase, or deferred until the Validation Phase, but the procedure shall be written prior to commencing and Validation activity.

6.3.3 V&V Requirements for Acquired Software

Acquired software refers to:

1. support software (see SMP [2.2(5)], Support Software), such as commercially available software and development tools (i.e., compilers and databases),

2. third party software (i.e., software procured from outside sources and incorporated into the final software-based product), and

3. previously developed software.

The Responsible Technical Project Engineer shall assign engineer(s) with sufficient skill and knowledge to review the software documentation, to analyze the feasibility of using such software in software-based products (either as incorporated software or as software development tools) and to approve such use. If the acquired software contains Human-System Interfaces, an engineer who is knowledgeable in Human Factors Engineering (HFE) shall be involved in the evaluation process.

The evaluation process, at a minimum, shall be performed in accordance with the procedures outlined in the SMP [2.2(1)] and shall:

1. verify that the software has met the requirements for the intended use,

2. ensure that the engineering process used for previously developed software is equivalent to the engineering process outlined in the SMP [2.2(5)],

3. identify any additional activities to be performed (e.g., supplemental testing), and

4. identify all documents required for configuration control of the acquired software. The controlled documents, shall include, if applicable, the following information:

    a. user application requirements,

    b. test plan(s) and test cases required for software validation, and

    c. user interface documentation.

Documents such as the following may be useful references for performing support and/or third party software evaluation activities:

1. EPRI TR-106439, Guidelines on Evaluation and Acceptance of Commercial Grade Digital Equipment in Nuclear Safety Application [2.3.1(1)], and

2. NUREG/CR-6421, A Proposed Acceptance Process for Commercial Off-the-Shelf (COTS) Software in Reactor Applications [2.3.3(8)].

All acquired software requires an evaluation report, which shall be verified by the VVT in accordance with this plan, the SMP [2.2(1)] & SQAP [2.2(5)].

Verification need not be performed on the acquired software (as might be the case with a commercial compiler or other software development tools), if qualification can be achieved by validating the *output* of the software development tools. Such deviation is permissible if the output quality level is maintained and justification documented in the software project Design Record File (DRF). An evaluation report is still required.

Support software developed or used by the design team to produce design calculations, design bases, design data, or other data that affect the design, licensing, reliability, or operation of software-based products is beyond the scope of this plan, and is addressed by GEEN EOP 40-3.00, Engineering Computer Programs [2.2(7d)].

The evaluation and test results shall be verified and documented in the software project DRF.

The above documentation shall be assembled during the design definition phase and shall be subject to configuration management. The acquired software shall be placed under configuration control {see SCMP [2.2(2)]}. All documentation which cannot be stored in the Software Library {see SCMP [2.2(2)]} shall be filed in the software project DRF.

6.3.3.1 Support Software/Tool Documentation

The VVT shall examine the design organization's support software and tool documentation (e.g., commercially available support software or tools such as CASE) that supports the evaluation and acceptance of digital equipment for Nuclear Applications. The VVT shall verify the evaluation reports prepared by the design team to confirm:

1. Reliability, Testability, correctness, and accuracy
2. Risk has been assessed and minimized with the use of each support module or tool

6.3.3.2 3rd Party/Vendor Software Procurement Documentation

3rd party software can include libraries, functions, or vendor applications developed expressly for the project. The VVT shall examine the design organization's safety software procurement documentation for new or previously developed 3rd party/vendor software that supports the evaluation and acceptance of digital equipment for nuclear applications. The VVT will review the evaluation reports prepared by the design team to verify:

1. Consistency, accuracy, readability, security, and completeness
2. Technical and quality requirements, and identifying critical characteristics of commercial digital equipment to be used in systems are determined

3. Appropriate methods for verifying the critical characteristics, accepting digital products from commercial vendors and dedicating them for use in nuclear applications are identified

4. The dedication (for Quality Class Q applications) basis is maintained to ensure that it remains valid over the operating lifetime of the equipment in the plant

6.3.3.3 Previously Developed Software Documentation

Previously developed software is software developed by GEEN prior to the project. The VVT shall independently verify:

1. Previously Developed Software Evaluation Report prepared by the design team

2. Previously Developed Software Supplemental Documentation

To confirm:

1. Consistency, accuracy, readability, and completeness

2. Any supplemental development or modification to the previously developed software for the project has been done in accordance to applicable plans, procedures or processes (e.g., development for Quality Class Q systems shall follow the SMP and other applicable plans)

3. Appropriate methods have been identified, and instituted by the design team, (e.g., auditing, inspection or independent review) for verifying the original design remains intact and that critical characteristics of the software have been properly considered for use in digital products, from commercial vendors or from GEEN, especially for systems that will be dedicated for use in nuclear safety applications.

6.3.3.4 Commercial Off-The-Shelf (COTS) Software Documentation

Commercial off-the-shelf software can include libraries, functions or complete applications that are commercially available to anyone and are used in the project software applications. The VVT shall independently verify:

1. COTS Report prepared by the design team

2. COTS Supplemental Documentation to confirm:

    a. Technical, security and quality requirements, and identifying critical characteristics of commercial digital equipment to be used in safety systems are determined,

    b. Appropriate methods for verifying the critical characteristics, accepting digital products from commercial vendors and dedicating them for use in nuclear safety applications are identified, and

c. The dedication basis is maintained to ensure that it remains valid over the operating lifetime of the equipment in the plant.

The examination shall be based on EPRI-TR-106439, Guidelines on Evaluation and Acceptance of Commercial Grade Digital Equipment in Nuclear Safety Applications [2.3.1(1)].

6.3.3.5 Embedded Software

Software that is not directly part of the project, and run on the distributed control and information system (DCIS), but is accessed by and can provide a digital interface directly to the user and or the DCIS is considered as embedded by this plan.

The VVT shall independently review documentation of all Class Q (safety) digital computer systems (including systems containing embedded software) used for the project DCIS to confirm that the system has been qualified to the Appendix B QA program. Furthermore the VVT shall inspect applicable Appendix B documentation for consistency, completeness, reliability, accuracy and requirements traceability (both forward and backwards). Digital computer systems not adhering to the Appendix B QA shall adhere to the appropriate guidelines of this SVVP (e.g., vendor, 3rd party, COTS, or in-house software).

6.3.4 Requirements Phase Safety Analysis

The Software Safety Plan [2.2(4)] describes the software safety analysis to be performed by the Design Team for each software life cycle phase. The Software Safety Team (SST) shall verify design definition phase safety analysis documentation to confirm:

1. Design definition phase safety analysis implementation tasks have been carried out in their entirety.
2. System safety and security requirements are correctly addressed.
3. No new hazards have been introduced.
4. Software requirements that can affect safety are identified.
5. All other software requirements, design elements, and code elements do not affect safety.

6.3.5 Requirements Traceability Analysis

The VVT shall examine the requirements traceability matrix (or equivalent traceability documentation) of Quality Class Q software to confirm:

1. Software requirements documented in the SRS or the appropriate document(s) are traceable to system requirements and system design documents.
2. The matrix allows traceability in both a forward and backward direction.

3. Identified relationships are correct, consistent, complete, and accurate.

6.3.6 Software Requirements

The VVT shall examine the Software Requirement Specification (SRS) to verify that:

1. The SRS is complete, consistent, correct, accurate and testable.
2. The requirements comply with established standards, practices and conventions.
3. The SRS satisfies system software objectives.
4. The SRS addresses and identifies critical requirements key to software performance.
5. the SRS or software requirements in IPS, for Quality Class Q, software meets the traceability requirements in Reg. Guide 1.172[1.3.2.5(3)]

6.3.7 Software Requirements Interface Analysis

The VVT shall verify the Software Requirement Specification (SRS) with hardware, operator, user, and software interface requirements to confirm completeness, accuracy and readability and that the specified software and interface requirements have been evaluated and are traceable to the requirements as specified in the input documents and applicable requirement standards

6.3.8 System Test Plan

The VVT shall verify the system test plan to assure testing demonstrates that software satisfies system objectives by confirming that:

1. The system test plan for Quality Class Q Software is traceable from system test requirements to test design, test cases, test procedures, and test results.
2. Planned testing demonstrates compliance with all functional requirements as a final software product in the system environment.
3. Planned testing demonstrates system performance at hardware, software, user, and operator interfaces.
4. Planned testing includes demonstration of the adequacy of user documentation.
5. Planned testing demonstrates security attributes, and system performance at boundaries (for example, data, interface) and under stress conditions.

6.3.9 Acceptance Test Plan

The VVT shall verify the acceptance test plan to assure testing demonstrates that software correctly implement system and software requirements in an operational environment by confirming that:

1. Planned testing demonstrates compliance with acceptance requirements in operational environment

2. Planned testing includes demonstration of the adequacy of user documentation.

3. Planned tests demonstrate traceability of acceptance test requirements to test design, cases, procedures, and results

6.3.10 Software Architecture Description

The VVT shall verify the Software Architecture Description to confirm correctness, consistency, completeness, accuracy, readability, and testability as well as overall review of the architecture description tasks and results.

6.4 Design Phase

V&V activities for the software design phase of the project shall include tasks as described in the following sections. The described task should be considered as a minimum and the VVT shall have the authority to expand these tasks at its discretion.

6.4.1 Design Safety Analysis

The Software Safety Plan describes the software safety analysis to be performed by the Design Team for each software life cycle phase. The SST shall verify the design phase safety analysis documentation to confirm:

1. Design phase safety analysis implementation tasks have been carried out in their entirety and in accordance with Software Safety Plan (SSP) [2.2(4)].

2. System safety requirements are correctly addressed.

3. No new unresolved hazards have been introduced.

4. Software requirements and design elements that can affect safety are identified.

5. All other software requirements and design elements do not affect safety.

6.4.2 Design Traceability Analysis

Traceability requires there to be a two-way trace (i.e., backward and forward traceability) between the software design requirements defined in the SRS and software design elements of the SDS. The VVT shall verify the design traceability matrix (or equivalent traceability documentation) for Quality Class Q Software to confirm:

1. Software Design Specification (SDS) requirements are traceable to Software Requirements Specification (SRS).

2. The matrix allows traceability in both directions.

3. Identified relationships are correct, consistent, complete, and accurate.

4. That the software design specification requirements are also traceable to the software architecture description.

6.4.3 Design

The VVT shall verify the SDS for overall consistency, completeness, accuracy and testability and that the software requirements are configured into well structured modules, and are traceable to the input documents.

The VVT shall also review SDS documentation to see that it is compliant to applicable IEEE and NUREG standards and that the software requirements are correctly implemented, and that the design is consistent, accurate and testable with respect to the requirements. They comply with established standards, practices and conventions and address and identify critical requirements key to software performance.

6.4.4 Design Interface Analysis

The VVT shall verify the SDS with hardware, operator, and software interface requirements in order to confirm:

1. Correctness, consistency, completeness, and accuracy

2. The data items at each interface are analyzed

6.4.5 Component Test Plan

The VVT shall verify the component test plan to assure testing demonstrates that software satisfies system objectives by confirming that:

1. Software elements (for example, units, modules) correctly implement component requirements.

2. There is traceability of design requirements for Quality Class Q software to test design, cases, procedures, and execution results.

3. Test termination criteria (e.g., test coverage and definition of success) are clearly defined and are adequate for the test.

6.4.6 Integration Test Plan

The VVT shall verify the Integration test plan to assure testing demonstrates that software satisfies system objectives by confirming that:

1. Software (for example, sub-elements, interfaces) correctly implements the software requirements and design.

2. There is traceability of requirements for Quality Class Q software to test design, cases, procedures, and execution results.

3. Test termination criteria (e.g., test coverage and definition of success) are clearly defined and are adequate for the test.

6.4.7 Test Design

The VVT shall verify the test design documentation in order to confirm:

1. The test approach for software features and combination of software features is adequate for:

   a. Component testing

   b. Integration testing

   c. System testing

   d. Acceptance testing

2. There is traceability of test design documentation for Quality Class Q software as required by the Test Plan

3. Test termination criteria (e.g., test coverage and definition of success) are clearly defined and are adequate for the test

6.4.8 User's Manual & Documentation

The VVT shall perform a verification of the complete User's Manual and any other documents generated for explicit use by the user (e.g., O&MM - Operations and Maintenance Manual) during development to confirm:

1. Correctness

2. Consistency

3. Completeness

4. Accuracy

5. Readability

6. Testability

7. Traceability to system and software requirements

8. Security

Operations and Maintenance Manuals shall provide user information to operate and maintain the software and associated equipment, including installation and configuration tables.

6.4.9 Software Code Style Guide

The VVT shall verify the software coding style guide to confirm it meets the applicable standards and guidelines.

6.4.10 Validation Test Procedures

The VVT shall verify the Validation Test Procedures and Test Cases Specification contain a complete set of test conditions such as,

1. testing performed on all hardware, software and user interfaces, and
2. testing performed at abnormal and stress conditions (i.e., that such testing, including the acceptance criteria, is defined for each test case),

Also verify validation test procedures for Testability, readability, consistency, completeness and accuracy and traceability (for Quality Class Q software) (forward and backwards) of test criteria to software requirements (function, informational and behavioral).

6.4.11 Design Metrics

Design metrics shall be collected and utilized by the SPE as described in the SQAP [2.2(5)].

6.5 Implementation Phase

V&V activity for the implementation phase of the project shall include tasks as described in the following sections. The described task should be considered as a minimum and the VVT shall have the authority to expand these tasks at its discretion. The overall goal of the VVT with regards to the implementation phase is to verify and validate high-level design output as well as to verify and validate that the design teams design V&V is complete, consistent, accurate and reliable. This shall include all aspects of design V&V and the VVT shall have the authority, at its discretion, to be involved in design V&V activities in the manner that the VVT can assure the integrity of the software and related processes.

6.5.1 Implementation Safety Analysis

The Software Safety Plan describes the software safety analysis to be performed by the Design Team for each software life cycle phase. The Software Safety Team (SST) shall verify implementation phase safety analysis documentation to confirm:

1. Implementation phase safety analysis implementation tasks have been carried out in their entirety
2. System safety requirements are correctly addressed
3. No new hazards have been introduced

4. Software requirements, design elements, and code elements that can affect safety are identified

5. All other software requirements, design elements, and code elements do not affect safety

6.5.2 Source Code Traceability Analysis

Traceability requires there to be a two-way trace (i.e., backward and forward traceability) between the elements of the detailed design and the elements in the code. The VVT shall examine the source code traceability matrix (or equivalent traceability documentation) of Quality Class Q software to confirm:

1. Source code traceability to corresponding design specification(s) and design specification(s) to source code

2. Identified relationships are correct, consistent, complete, and accurate

6.5.3 Source Code

The VVT shall examine and verify, all source code (in the form of code reviews) in order to confirm:

1. The code correctly implements the design requirements specified in the SDS, and

2. The code complies with the guidelines outlined in the Software Conventions and Guidelines document.

3. Correctness, consistency, completeness, accuracy, testability and security

4. Compliance with established standards, practices and conventions

5. Quality of the source code

Code reviews shall be performed as outlined in the SMP [2.2(1)] and SQAP [2.2(5)].

The results of the code reviews, any discrepancies found, and all review comments shall be documented in individual code review data sheets and included as part of the Module Test Report.

6.5.4 Source Code Interface Analysis

As part of the code reviews, the VVT shall verify the source code interfaces in order to confirm:

1. Source code with hardware, operator, and software interface design documentation is correct, consistent, complete, and accurate.

2. Analysis considers the data items at each interface.

6.5.5 Source Code Documentation

As part of the code reviews, the VVT shall verify source code documentation in order to confirm that:

1. Source code documentation with source code is correct, consistent, and complete.

6.5.6 Module Testing

Module testing is performed to:

1. confirm the performance of the individual software modules of the software-based product,
2. verify that the software implemented is consistent with the requirements specified in the SDS, and
3. evaluate software module interfaces (including interfaces with the hardware and the user), local data structures, normal processing paths, error paths, and boundary conditions.

Module testing shall be performed as outlined in the SMP [2.2(1)], SQAP [2.2(5)] and Software Test Plan. The extent of the module testing performed and the test results obtained shall be documented in individual module test data sheets and included as part of the Module Test Report.

6.5.7 Test Procedure (Module Testing)

The VVT shall verify the all module test procedure documentation in order to confirm:

1. A sequence of actions for the execution of tests has been specified and that the sequence is technically adequate to demonstrate all requirements have been met.
2. There is traceability of test procedure documentation as required by the Test Plan.
3. Adherence to Table 4.

6.5.8 Module Test Report

The Module Test Report shall be generated to report:

1. the results of code review, review comments and resolution to these comments,
2. the test steps executed for each module test,
3. the results of each module test, test comments and resolution to these comments,
4. the test methods and tools used in each module test, and
5. the problems encountered during testing.

V&V shall be performed on Module Test Report. The Responsible Verifier shall ensure that:

1. the design requirements specified in the SDS are correctly implemented,
2. the testing is performed in accordance with the methods outlined in the Software Test Plan (i.e., test methods and test cases used), and
3. all discrepancies found are addressed and resolved by the RE via comments documented on the code review data sheets and module test data sheets.
4. test cases are traceable to appropriate design documentation,
5. all source code files are controlled as configurable items, and each test case can be traced to a controlled revision.
6. A sequence of actions for the execution of tests has been specified and that the sequence is technically adequate to demonstrate all requirements have been met,
7. All test were carried out according to the plan and procedures,
8. module testing error checklist are documented and tracked,
9. the report is correct, consistent, complete, and accurate.

6.5.9 Implementation Phase Metrics

The VVT shall collect and utilize metrics specific to the implementation phase as specified by the SQAP [2.2(5)].

6.6 Integration Phase

The VVT shall verify all documents associated with testing activities and shall at its discretion, participate in all testing activities at various levels. Principle responsibility for performing the actual testing remains with the design team.

6.6.1 Test Safety Analysis

The Software Safety Plan describes the software safety analysis to be performed by the Design Team for each software life cycle phase. The Software Safety Team (SST) shall verify test phase safety analysis documentation to confirm:

1. Test phase safety analysis implementation tasks have been carried out in their entirety.
2. System safety requirements are correctly addressed.
3. No new hazards have been introduced.

4. Software requirements, design elements, and code elements that can affect safety are identified.

5. All other software requirements, design elements, and code elements do not affect safety.

6.6.2 Integration Testing

Table 7 lists the design outputs, their verification documents and the V&V output for the Integration Test Phase. Integration testing is performed as outlined in the SMP [2.2(1)], SQAP [2.2(5)] and Software Test Plan to ensure that:

1. the integrated software modules performed as intended,

2. the software, when installed in the prototype or target hardware, performs as intended,

3. interfacing entities as specified, in HSS and SRS for software-based systems, and HSS, SBD, and IPS for microprocessor-based systems, operate as intended,

4. the self-test functions perform correctly, and

5. that the data communication functions perform correctly,

Integration testing shall involve the use of software emulation tools to:

1. check internal results or intermediate variables,

2. create the conditions required to confirm limit control, and

3. confirm system performance that cannot be directly measured from the external inputs and outputs, and

4. for Quality Class Q software, test the inter-module logic paths.

Integration Test Data Sheets shall be generated to document the:

1. test objectives and test results,

2. test procedures executed,

3. test methods and equipment/tools used,

4. problems encountered.

These test data sheets shall be included as part of the Integration and Installation Test Report. V&V shall be performed on Integration and Installation Test Report and traceability analysis report. The Responsible Verifier shall ensure that:

1. the software-based product meets the requirements specified in the HSS and SRS for software-based systems and in the HSS, SBD and IPS for microprocessor-based systems,
2. any discrepancies documented on the test data sheets have been resolved by the RE, and
3. software errors as defined in the Error Checklist {see SMP [2.2(1)]} are tracked and documented.
4. all source code files are controlled as configurable items, and each test cases can be traced to a controlled revision.
5. software and hardware (if applicable) satisfies acceptance criteria.
6. There is traceability of acceptance criteria documentation as required by the Test Plan.
7. All test were carried out according to the plan and procedures.
8. All anomalies have been resolved and closed out.
9. the report is correct, consistent, complete, and accurate.

Problems requiring code changes shall be documented in the Integration Test Data Sheets, along with the justification for the extent of additional module testing and review to be performed.

6.6.3 System Build Documentation

The VVT shall verify the System Build Documentation in order to confirm completeness, correctness, consistency and accuracy.

6.6.4 Integration Test Metrics

The VVT shall collect and utilize metrics specific to the software-coding phase as specified by the SQAP [2.2(5)].

6.7 Validation Test Phase

V&V activity for the validation test phase of the project shall include tasks as described in the following sections. The overall goal of the VVT with regards to the validation test phase is to validate aspects of the testing output as well as to verify and validate that the design team's design is complete, consistent, accurate and reliable.

Table 8 lists the design outputs, their verification documents and the V&V output for the Validation Test Phase. Validation testing shall be conducted according to the SQAP [2.2(5)],

Software Test Plan [2.2(6)] and Validation Test Procedures and Test Cases Specification {see SMP[2.2(1)]}.

The validation test may be performed on either production hardware or prototypical hardware (hardware that has been confirmed to be fully functionally equivalent). Validation testing shall be performed by individuals other than the (system/instrument) software designers. All test tools used shall be documented in the Validation Test Report, and their calibration shall be current.

The purpose of validation testing is to confirm:

1. that the software-based product is operational and conforms to all functional and performance requirements contained in the HSS, and

2. that the HSI functions implemented are consistent with the final version of the User's Manual.

3. verify and validate the build and installation documentation (i.e. Build Release Description)

The results of the validation test, including anomalies discovered and their resolution, and test summary shall be documented in a Validation Test Report. V&V shall be performed on the Validation Test Report to verify that it is Correct, consistent, complete, accurate, readable and testable.

6.7.1 Validation Safety Analysis

The SST shall verify all Validation Safety Analysis packages to confirm:

1. Correctness, consistency, completeness, accuracy, readability and testability.

2. Adherence to Table 4.

6.7.2 Validation Test Plan

The VVT shall verify the Validation Test Plan to confirm:

1. Correctness, consistency, completeness, accuracy, readability and testability.

2. Adherence to Table 4.

6.7.3 Validation Test Report

The VVT shall verify the Validation Test Report to confirm:

1. Correctness, consistency, completeness, accuracy, readability and testability.

2. Adherence to Table 4.

6.7.4 Validation Test Metrics

The VVT shall collect and utilize metrics specific to the validation phase as specified by the SQAP [2.2(5)].

6.8 Installation Phase

Physical installation activities are outside the scope of this document. Requirements for Installation Phase activities are contract dependant and as such cannot be defined in the scope of this document.

6.9 Operations and Maintenance Phase

Operation and maintenance phase activities are outside the scope of this document. Requirements for O&M Phase activities are contract dependant and as such cannot be defined in the scope of this document.

6.10 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. Such reviews evaluate the design outputs (including V&V and software safety analysis outputs) for conformance to the design process, and completeness of the software engineering process. The Baseline Review process is under the control of the Software Project Engineering Organization. Baseline Reviews are to be performed by qualified persons other than those directly responsible for the design. These reviews are used to confirm adherence to the SMP [2.2(1)], SQAP [2.2(5)], SCMP [2.2(2)], and SVVP (this plan), and SSP [2.2(4)]. The results of all baseline reviews shall be documented in a Baseline Review Record. The Baseline Review Record summarizes the design, configuration management, software safety analysis and V&V activities of the particular software life cycle phase. The BRR shall be filed in the software project DRF. The outputs from the baseline reviews shall be maintained under configuration management according to the process specified in the SCMP [2.2(2)].

The specific baseline reviews to be performed, as defined in the SMP [2.2(1)] are listed below.

6.10.1 Planning Phase

The purpose of the this baseline review is to confirm:

1. that the scope for the software-based product has been adequately defined,

2. that V&V and are completed on the design outputs (see Table 3) and equipment design requirements (e.g., SDD and Logic Diagrams), and all contractual technical agreements have been incorporated and are ready to serve as the input documents for the next software life cycle phase,

3. the completion status of the design outputs, and

4. that the resources and standards to be used on the software-based product have been identified and properly documented.

The Planning Baseline Review Record shall be used to document any anomalies or nonconformances identified during the Baseline Review, and shall contain a specific statement of the review results. The RE shall resolve each anomaly or nonconformance for the Configuration Items (including the review report) before the software design and development effort can be formally closed. The SMP [2.2(1)], SQAP [2.2(5)], SCMP [2.2(2)], and SVVP (this plan) shall be the governing documents of the software engineering process.

6.10.2 Requirements Phase

The purpose of the requirements baseline review is to confirm:

1. that the required design definition activities, such as traceability analysis and verification are completed on the design outputs, and

2. the completion status of the design outputs.

The Requirements Baseline Review Record shall be used to document any anomalies or nonconformances identified during the Baseline Review and shall contain a specific statement of the review results. Each anomaly or nonconformance shall be resolved by the RE for the Configuration Items (including the review report) before the software design and development effort can be formally closed.

6.10.3 Design Phase

The purpose of the design baseline review is to confirm:

1. that the required software design activities, such as traceability analysis and verification are completed on all design outputs, and

2. the completion status of the design outputs.

The Design Baseline Review Record shall be used to document any anomalies or nonconformances identified during the Baseline Review and shall contain a specific statement of the review results. The RE shall resolve each anomaly or nonconformance for the Configuration Items (including the review report) before the software design and development effort can be formally closed.

6.10.4 Implementation Phase

The purpose of the implementation baseline review is to confirm:

1. that the required Software Coding activities, such as code review and module test are completed,

2. that the required traceability analysis and verification are completed on the design outputs, and

3. the completion status of the design outputs.

The Implementation Baseline Review Record shall be used to document any anomalies or nonconformances identified during the Baseline Review and shall contain a specific statement of the review results. The RE shall resolve each anomaly or nonconformance for the Configuration Items (including the review report) before the software design and development effort can be formally closed.

6.10.5 Integration Phase

The purpose of the integration baseline review is to confirm:

1. that the required integration test activities are completed,

2. that the required traceability analysis and verification are completed on the design outputs, and

3. the completion status of the design outputs.

The Integration Baseline Review Record shall be used to document any anomalies or nonconformances identified during the Baseline Review and shall contain a specific statement of the review results. The RE shall resolve each anomaly or nonconformance for the Configuration Items (including the review report) before the software design and development effort can be formally closed.

6.10.6 Validation Test Phase

The purpose of the validation test baseline review is to confirm:

1. that the required validation test activities are completed,

2. that the required traceability analysis and verification are completed on the design outputs, and

3. the completion status of the design outputs.

Physical and Functional Configuration Audits {see SCMP [2.2(2)]} will be performed as part of the Validation Test Baseline Review to ensure that:

1. the completeness of the System Build Description, and

2. the software-based product has satisfied the requirements or functions defined in the HSS.

7 Software Verification and Validation Reporting

All V&V activities culminate with reporting as the final objective. The style and content expected in such reports is described in the next three sections and are segregated according to the final report, anomaly reports and management reports.

7.1 V&V Reporting

V&V team members shall prepare an Independent V&V Report for each VVT activity performed. All verification reporting shall be done in accordance with EOP 40-6.2 [2.2(7b)].

Validation reports are shall be done in accordance with the requirements for appropriate phase test reports as specified in the SMP [2.2(1)], STP [2.2(6)], SVVP (this plan) & the SQAP [2.2(5)].

7.2 Anomaly Reporting

An anomaly is anything observed in the documentation or operation of software that deviates from expectations based on previous verified software products or reference documents. Anomalies discovered during an V&V activity shall be documented either in the verification report as specified by the GEEN EOP, or in the test report as specified by the SMP [2.2(1)], STP [2.2(6)] & SVVP (this plan).

7.3 Engineering Management Report

V&V team members, under the direction of the VVT task lead, shall periodically (when needed) and at the end of a phase, prepare Engineering Management Reports to summarize current V&V activities and status. Such reports shall include, but are not limited to:

1. A list of open items or unresolved anomalies, which includes the responsible engineer and date item was open and date expected to close item. In the case of a Phase summary a list of all anomalies and their resolution should also be included.

2. Specific milestones, summaries and descriptions of V&V tasks performed as well as the task results (e.g., completion of a phase baseline or documents or activities the VVT has completed since the last report.)

3. Any recommendations.

Such reports will be submitted to the Engineering management to assist in communicating open V&V items and to assist in the timely resolution of such items or in the case of a the end of a phase to document it as such.

7.4 Final V&V Final Report

The VVT shall summarize all V&V activities and results, including status and disposition of anomalies in an Independent V&V final report. The report shall include, at a minimum, applicable items in a normal V&V report as well as:

1. Summary of all life cycle V&V tasks.
2. Summary of task results.
3. Summary of anomalies and resolutions (for Quality Class Q software only).
4. Assessment of overall software quality.
5. Any Final V&V recommendations

8 Verification and Validation Administrative Procedures

Verification and Validation Administrative Procedures conjoined to the V&V activities are outlined in the following sections. Such procedures explain the manner in which the VVT will perform activities such as reporting and task iteration that are outlined in this V&V plan.

8.1 VVT Schedule

The design team shall inform the VVT of the list of documents for verification for each software development phase. A reasonable time (i.e., two months) should be determined for submittal of documents to the VVT and can vary depending on the phase or product. Submittal shall be prior to the scheduled baseline review date. The VVT task lead shall arrange for VVT review and inform V&V Team Members of their assigned responsibilities and schedule prior to completion date. Such scheduling should be a reasonable amount of time (i.e., one month), which can vary with phase or product size.

8.2 Nonconformance & Anomaly Reporting and Resolution

Anomalies discovered during Independent V&V are documented as described in Section 7.2, "Anomaly Report."

Nonconformance and anomaly comments logged as part of the V&V process shall be addressed and resolved by the RE and maintained in the software project Design Record File. Section 6.9 describes the documentation methods for nonconformance or discrepancies identified during V&V activities.

If the identified nonconformance or discrepancy impacts a baselined configuration item, the change control process defined in the SCMP [2.2(2)] shall be followed.

All nonconformance comments generated, as part of the V&V process, shall be resolved before the verification effort can be considered complete. The status and the adequacy of V&V performed will be documented in the Baseline Review Record, see SCMP [2.2(2)].

8.3 Task Iteration Policy

When changes are made to the input of a base-lined software life cycle phase, an assessment shall be made of the extent that affected V&V tasks shall be re-performed by the Technical Project Engineer.

Any changes made to the software after validation testing has been completed shall be handled in accordance with the SCMP {see SCMP [2.2(2)]}. V&V shall be performed on all revised documentation. The RTPE shall determine the supplemental V&V that is required to ensure that no errors are induced as the results of the change activity.

8.4 Deviation Policy

Where deviation from the SVVP is deemed necessary, a request shall be presented to the SPE Task Lead for review and approval. The Deviation Request shall include:

1. Identification of the V&V task to be amended or deleted.
2. Rationale for the requested change.
3. Assessment of the impact on the software quality.

A copy of the approved Deviation Request shall be included in the V&V Review Report.

8.5 Control Procedures

All records of VVT software V&V activities shall be permanently retained in the appropriate DRF. This includes:

1. V&V Review Reports
2. Anomaly Reports
3. Evidence of anomaly resolution, for example, baseline review reports
4. Approved SVVP Deviation Requests
5. Management Review Reports

The VVT task lead is responsible for placing SPE records into the DRF.

The procedures for how software products are to be configured, protected, and stored are beyond the scope of the SVVP. These requirements are specified in the project SCMP [2.2(2)].

The VVT has a responsibility to assure that Configuration Management (CM) outlined in the DCIS SCMP is controlled by this procedure and any outlined in the CM. The GEEN QA organization is an oversight organization that can, at its discretion, audit VVT activities to see that the VVT is maintaining control as described above.

8.6 Standards, Practices, and Conventions

The standards, practices, and conventions that govern this plan are contained in the References (see Section 2).

Table 1 Life Cycle Phase Cross-Reference

BTP HICB-14	IEEE 1012	NUREG/CR-6101	GE SMP
• Planning	• Concept	• Planning	• Planning
• Requirements	• Requirements	• Requirements	• Requirements
• Design	• Design	• Design	• Design
• Implementation	• Implementation	• Implementation	• Implementation
• Integration	• Test	• Integration	• Integration
• Validation	• Test	• Validation	• Validation
• Installation	• Installation and Checkout	• Installation	• Installation
• Operation and Maintenance	• Operation and Maintenance		• Operation and Maintenance

Table 2 SVVP Cross-Reference to SRP Section 7, BTP HICB-14 [2.3.3(3)]

BTP HICB-14 Requirement		SVVP Section
B.2	Information to be Reviewed	N/A
B.2.1	Software Life Cycle Process Planning	6.2.1
B.2.2	Software Life Cycle Process Implementation	N/A
	Safety analyses	6.3.4, 6.4.1, 6.5.1, 6.6.1, 6.71
	Verification and validation analysis and test reports	Table 4
	Configuration management reports	BRT performs CM function, 6.9
B.2.3	Software Life Cycle Process Design Outputs	N/A
	Software requirements specifications (SRS)	6.3.1
	Hardware and software architecture descriptions (SAD)	6.3.10
	Software design specifications (SDS)	6.4.3, 6.4.4
	Code listings	6.5.3
	Build documents	6.6.3
	Installation and configuration tables	6.4.8
	Operations manuals	6.4.8
	Maintenance manuals	6.48
	Training manuals	Not Applicable
B.3	Acceptance Criteria	N/A
B.3.1	Software Life Cycle Process Planning	Not Applicable to this document (criteria in this section is that which is specified by Table 4 and is used by this SVVP, but is not criteria for this SVVP)
B.3.2	Software Life Cycle Process Implementation	N/A
B.3.2.a	Safety Analysis Activities	N/A
	Safety analysis performed for each life cycle group, safety analysis items confirmed	6.3.9, 6.4.1, 6.5.1, 6.6.1, 6.7.1

Table 2 SVVP Cross-Reference to SRP Section 7, BTP HICB-14 [2.3.3(3)] (Continued)

BTP HICB-14 Requirement		SVVP Section
B.3.2.b	Software Verification and Validation Activities	N/A
	V&V report produced for each life cycle group. Documentation shows that requirements, design, code, integration, and installation outputs satisfy functional and process characteristics.	Table 4. Details specified in 6.3, 6.4, 6.5, 6.6, 6.7
	Verification problems and mitigating actions are recorded	5.3
	Design documents are traceable to system requirements	6.3.5, 6.4.2, 6.5.2
	Integration V&V activities have the required attributes	6.4.6, 6.5.7, 6.6.3
	Software validation activities have the required attributes	6.3.2, 6.5.7
	Installation (acceptance) test activities have the required attributes	Table 4
B.3.2.c	Software Configuration Management Activities	N/A
	Documentation under configuration management control.	8.3
	Change control process identified.	8.6
B.3.3	Software Life Cycle Process Design Outputs	N/A
B.3.3.a	Requirements Activities - Software Requirements Specification	6.3.1
B.3.3.b	Design Activities - Software Architecture Description	6.3.10
B.3.3.c	Design Activities - Software Design Specification	6.4.3, 6.4.4
B.3.3.d	Implementation Activities - Code Listings	6.5.3, 6.5.4, 6.5.5
B.3.3.e	Integration Activities - System Build Documents	6.6.3
B.3.3.f	Integration Activities - Installation Configuration Tables	6.7.1
B.3.3.g	Installation Activities - Operations Manuals	6.4.8
B.3.3.h	Installation Activities – Maintenance Manuals	6.4.8
B.3.3.i	Installation Activities – Training Manuals	Not Applicable to this document

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)]

IEEE 1012 Requirement		SVVP Section, Paragraph
3.1	Purpose	Sec 1
	Purpose delineated	Sec 1.2, Application & Sec 1, Purpose
	Scope delineated	Sec 1.1, Scope
	Waivers delineated	Sec 1.2, Application, Para. 1
	Project identified	Sec 1, Para 1, "DCIS division of the Project"
	V&V goals specified	Sec 1, Para 3, list of objectives
3.2	Referenced Documents	Sec 2
	Binding compliance documents identified	Sec 2.3 Codes and Standards, "codes and standards are applicable ... to the extent specified in this plan"
	Reference documents identified	Sec 2.2, Supplemental Documents
	Required supporting documents identified	Sec 2.1 Supporting Documents
3.3	Definitions	Sec 3
	Define or provide reference of all terms required to implement the Plan	Sec 3.2, Definitions
	Describe acronyms and notations	Sec 3.1, Acronyms
3.4	Verification and Validation Overview	Sec 4.
3.4.1	Organization	Sec 4.1
	Describe organization	Sec 4.1
	Define the relationship to other V&V efforts	Sec 4.2
	Define lines of communication	Sec 4.2, Figure 1
	Define authority for resolving issues	Sec 4.1, Para 2
	Define authority for approving V&V products	Sec 4.1, Para 1 & Figure 2

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement		SVVP Section, Paragraph
3.4.2	Master Schedule	4.3
	Describe project life cycle	Sec 3.2.1
	Describe milestones	DCIS ECS DPP
	Summarize scheduling of V&V tasks	DCIS ECS DPP
	Describe V&V result feedback to development process	Sec 4.2
	If Plan life cycle differs from standard life cycle, show how standard requirements are satisfied	Table 3 (This table)
3.4.3	Resources Summary	Section 4.4
	Staffing	Sec. 4.4
	Facilities	N/A
	Tools	Special tools are handled in Section 3.4.5 of this table.
	Finances	N/A (Finances are beyond the scope of this Plan)
	Special procedural requirements (security, access rights, documentation control)	N/A
3.4.4	Responsibilities	Section 4.5
	Identify organizational element(s) for each V&V task	Sec 4.5
	Identify specific responsibility of each element for tasks assigned to more than one element	Sec 4.5
3.4.5	Tools, Techniques, and Methodologies	Sec 4.6
	Identify, describe purpose and use of, and plans for acquisition, training, support, and qualification of special software tools used for V&V.	Sec 4.6.7
	Identify, describe purpose and use of, and plans for acquisition, training, support, and qualification of special techniques used for V&V.	Sec 4.6.1, 4.6.2, 4.6.3
	Identify, describe purpose and use of, and plans for acquisition, training, support, and qualification of special methodologies used for V&V.	Sec 4.6.1, 4.6.2, 4.6.3
3.5.1	Management of V&V	Sec 5
	Phase Independent Requirements	*N/A*
	(1) Verification and Validation Tasks	Sec 4.7

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
Identify independent V&V tasks for each phase	Table 4
Describe how each task contributes to independent V&V goals	Sec 4.6, 5.1, 5.2, 5.3, 6.2.1, 6.2.2, 6.3.1, 6.3.2, 6.3.3., 6.3.4., 6.3.5., 6.3.6., 6.3.7., 6.3.8., 6.3.9, 6.3.10, 6.4.1., 6.4.2, 6.4.3, 6.4.4, 6.4.5, 6.4.6, 6.4.7, 6.4.8, 6.4.9, 6.4.10, 6.4.11, 6.5.1, 6.5.2, 6.5.3, 6.5.4, 6.5.6, 6.5.7, 6.5.8, 6.5.9, 6.6.1, 6.6.2, 6.6.3, 6.6.4, 6.7.1, 6.7.2, 6.7.3, 6.7.4, 6.8, 6.9.1, 6.9.2, 6.9.3, 6.9.4, 6.9.5, 6.9.6
(2) Methods and Criteria	Table 4
Identify methods and criteria	Table 4
Describe specific methods and procedures for each task	Table 4
Define detailed criteria for evaluating task results	Table 4, Sec. 6.1.3
(3) Inputs/Outputs	Table 4
Identify inputs required for each task	Table 4
Specify source and format of each input	Table 4
Identify output from each task	Table 4
Specify purpose and format for each output	Table 4
Each critical anomaly resolved before V&V proceeds to next phase	7.1, 7.2, 7.3
(4) Schedule	
Identify V&V task schedule	Sec 4.3, In Project Schedule
Establish milestones for initiating and completing task	In Project Schedule
Establish milestones for receipt of input	Sec 8.1
Establish milestones for delivery of output	In Project Schedule
(5) Resources	
Identify resources for performance of tasks	Sec 4.4
Specify resources by category	Sec 4.5
If tools are used, specify source of tools, availability, and other usage requirements	Sec 4.6.3, 8.3
(6) Risks and Assumptions	

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
Identify risks and assumptions associated with tasks including schedule, resources, or approach	Sec 5.3.1
Specify a contingency plan for each risk.	Sec 5.3.2
(7) Roles and Responsibilities	
Identify organizational elements of individuals responsible for tasks	Sec 4.5
Assign specific responsibilities for each task. (for each phase)	Sec 4.6, 5.1, 5.2, 5.3, 6.2.1, 6.2.2, 6.3.1, 6.3.2, 6.3.3., 6.3.4., 6.3.5., 6.3.6., 6.3.7., 6.3.8., 6.3.9, 6.3.10, 6.4.1., 6.4.2, 6.4.3, 6.4.4, 6.4.5, 6.4.6, 6.4.7, 6.4.8, 6.4.9, 6.4.10, 6.4.11, 6.5.1, 6.5.2, 6.5.3, 6.5.4, 6.5.6, 6.5.7, 6.5.8, 6.5.9, 6.6.1, 6.6.2, 6.6.3, 6.6.4, 6.7.1, 6.7.2, 6.7.3, 6.7.4, 6.8, 6.9.1, 6.9.2, 6.9.3, 6.9.4, 6.9.5, 6.9.6
Phase Dependent Requirements	*N/A*
Re-perform previous V&V tasks or initiate new V&V tasks to address software changes	Sec 8.3
V&V tasks are re-performed if errors are discovered in V&V inputs or outputs	Sec 8.3
(1) Software V&V Plan Generation	N/A VVT is not responsible for V&V Plan Generation
Generate SVVP (during Concept Phase) for all life cycle phases in accordance with this standard based on available documentation	N/A
Update SVVP for each life cycle phase	Subsec 4.5
Consider SVVP to be a living document – make changes as necessary	Subsec 4.5
Inputs: Development Schedules, Concept Documentation, SRS, Interface Requirements Documentation, SDD, Interface Design Documentation, Source Code Listing, Executable Code, User Documentation, Proposed Changes	Table 4
Output: SVVP, Updates	N/A
(2) Baseline Change Assessment	
Evaluate proposed software changes for effects on previously completed V&V tasks	Sec 8.4
Plan iteration of affected tasks	Sec 8.4
Input: Proposed Changes	Table 4.

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
Output: Updated SVVP	Table 4
(3) Management Review of V&V	
Conduct periodic reviews – V&V effort, technical accomplishments, resource utilization, future planning, and risk management	Sec 5.1 (1-5), 5.3.
Support daily management of V&V phase activities	Sec 5.1
Review task and V&V phase summary reports	Sec 7.3, 7.1
Evaluate V&V results and anomaly resolution	Sec 7.2
Input: Development Schedules, V&V Outputs	Table 4
Output: Task Reporting, V&V Phase Summary Reports	Table 4
(4) Review Support	
Correlate V&V task results	5.2
Identify key review support milestones	In Project Schedule
Schedule V&V tasks to meet milestones	Sec 5.2 Para 1
Establish methods to exchange V&V data and results	Sec 4.6 & 4.7
Input: V&V Outputs	Table 4
Output: Task Reporting, Anomaly Report	Table 4
3.5.2 Concept Phase V&V	
(1) Concept Documentation Evaluation	Sec 6.2
Evaluate concept documentation – satisfies user needs and project objectives	Sec 6.2.1
Identify major constraints of interfacing systems	Sec 6.2.1
Assess allocation of functions to hardware and software	Sec 6.2.1
Assess criticality of each software item	Sec 6.2.1
Input: Concept Documentation	Sec 6.2.1, Table 4.
Output: Task Reporting, Anomaly Report(s)	Sec 7, Table 4
Evaluate safety analysis documentation – satisfies user needs and project objectives	The preliminary Hazards Analysis provides this.
Identify major safety aspects of interfacing systems	Sec 6.2.2

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
Input: Concept Documentation, Safety Analysis	Table 4
Output: Task Reporting, Anomaly Report(s)	Sec 7, Table 4
3.5.3 Requirements Phase V&V	
(1) Software Requirements Traceability Analysis	Sec 6.3.5
Trace SRS requirements to system requirements in concept documentation	Sec 6.3.5 (1)
Analyze identified relationships for correctness, consistency, completeness, and accuracy	Sec 6.3.5(3)
Input: Concept Documentation, SRS Interface Requirements Documents	Sec 6.3
Output: Task Reporting, Anomaly Reports	Sec 7.2
(2) Software Requirements Evaluation	Sec 6.3.3
Evaluate SRS requirements for correctness, consistency, completeness, accuracy, readability, and testability	Sec 6.3.3 (1)
Assess how well SRS satisfies software system objectives	Sec 6.3.3 (3)
Assess the criticality of requirements to identify key performance or critical areas of software	Sec 6.3.3 (4)
Input: Concept Documentation, SRS, Interface Requirements Documentation	Table 4
Output: Task Reporting, Anomaly Reports	Table 4
(3) Software Requirements Interface Analysis	Sec 6.3.7
Evaluate SRS with hardware, user, operator, and software interface requirements documentation for correctness, consistency, completeness, accuracy, and readability	Sec 6.3.7
Input: SRS, Interface Requirements Documentation	Table 4
Output: Task Reporting, Anomaly Reports	Table 4
(4a) System Test Plan Generation - SVVP System Test Plan Evaluation – deviations noted	Sec 6.3.8
Plan system testing to determine if software satisfies system objectives	Sec 6.3.8 Para. 1
Criteria for determination:	N/A

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
(a) compliance with all functional requirements as complete software end item in system environment	Sec 6.3.8 (2)
(b) performance at hardware, software, user, and operator interfaces	Sec 6.3.8 (3)
(c) adequacy of user documentation	Sec 6.3.8 (4)
(d) performance at boundaries (for example, data, interface) and under stress conditions	Sec 6.3.8 (5)
Plan tracing of system end-item requirements to test design, cases, procedures, and execution results	Sec 6.3.8 (1)
Plan documentation of test tasks and results	Sec 4.6.4
Input: Concept Documentation, SRS, Interface Requirements Documentation, User Documentation	Sec 6.3
Output: System Test Plan, Anomaly Reports	Sec 7.2
(4b) Acceptance Test Plan Generation – SVVP Acceptance Test Plan Evaluation – deviations noted	Sec 6.3.9
Plan acceptance testing to determine if software correctly implements system and software requirements in an operational environment	Sec 6.3.9
Criteria for determination:	N/A
(a) compliance with acceptance requirements in operational environment	Sec 6.3.9 (1)
(b) adequacy of user documentation	Sec 6.3.9 (2)
Plan tracing of acceptance test requirements to test design, cases, procedures, and execution results	Sec 6.3.9 (3)
Plan documentation of test tasks and results	Sec 4.6.4
Input: Concept Documentation, SRS Interface Requirements Documentation, User Documentation	Sec 6.3
Output: Acceptance Test Plan, Anomaly Reports	Sec 7.2
Evaluate safety requirements analysis for correctness, consistency, completeness, accuracy, readability, and testability	Sec 6.3.4 (1)
Review plan documentation of analysis tasks and results	Sec 6.3.4
Input: Safety Analysis, SRS, Interface Requirements Documentation	Sec 6.4, Table 4

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
Output: Task Reporting, Anomaly Reports	Sec 7.2, Table 4
Evaluate user documentation for correctness, consistency, completeness, accuracy, readability, and testability	Sec 6.3.3
Input: SRS, Interface Requirements Documentation, User Documentation	Sec 6.3, Table 4
Output: Acceptance Test Plan, Anomaly Reports	Sec 7.2, Table 4
Evaluate user documentation for correctness, consistency, completeness, accuracy, readability, and testability	Sec 6.4.8
Review plan documentation of software architecture description tasks and results	Sec 6.4.8
Input: Software Architecture Description, Safety Analysis, SRS, Interface Requirements Documentation	Sec 6.3, 6.4, Table 4
Output: Acceptance Test Plan, Anomaly Reports	Sec 7.2, Table 4
3.5.4 Design Phase V&V	Sec 6.4
(1) Design Traceability Analysis	Sec 6.4.2
Trace SDD to SRS and SRS to SDD	Sec 6.4.2 (1)
Analyze identified relationships for correctness, consistency, completeness, and accuracy	Sec 6.4.2 (3)
Input: SRS, SDD, Interface Requirements Documentation, Interface Design Documentation	Sec 6.4, Table 4
Output: Task Reporting, Anomaly Reports	sec 7.2, Table 4
(2) Design Evaluation	Sec 6.4.3
Evaluate SDD for correctness, consistency, completeness, accuracy, and testability	Sec 6.4.3
Evaluate design for compliance with established standards, practices, and conventions	Sec 6.4.3
Assess design quality	Sec 6.4.3
Input: SDD, Interface Design Documentation, Standards (standards, practices, conventions)	Sec 6.4, Table 4.
Output: Task Reporting, Anomaly Reports	Sec. 7.2, Table 4

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
(3) Design Interface Analysis	Sec 6.4.4
Evaluate SDD with hardware, operator, and software interface requirements for correctness, consistency, completeness, and accuracy	Sec 6.4.4 (1)
Analyze data items at each interface	Sec 6.4.4. (2)
Input: SDD, Interface Design Documentation	Sec 6.4, Table 4
Output: Task Reporting, Anomaly Reports	Sec 7.2, Table 4
(4a) Component Test Plan Generation – SVVP Component Test Plan Evaluation – deviations noted	Sec 6.4.5
Plan component testing to determine if software elements (for example, unit, module) correctly implement component requirements	Sec 6.4.5. (1)
Criteria for determination:	Table 4
(a) compliance with design requirements	Table 4
(b) assessment of timing, sizing, accuracy	Table 4
(c) performance at boundaries and interfaces and under stress and error conditions	Table 4
(d) measures of test coverage and software reliability and maintainability	Table 4
Plan tracing of design requirements to test design, cases, procedures, and execution results	Sec 6.4.5 (2)
Review plan documentation of test tasks and results	Sec 6.4.5 (2)
Input: SRS, SDD, Interface Requirements Documentation, Interface Design Documentation	Sec 6.4, Table 4
Output: Component Test Plan, Anomaly Reports	Sec 7.2, Table 4
(4b) Integration Test Plan Generation – SVVP Integration Test Plan Evaluation – deviations noted	Sec 6.4.6
Plan integration testing to determine if software (for example, sub-elements, interfaces) correctly implements the software requirements and design	Sec 6.4.6 (1)
Criteria for determination:	N/A

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
(a) compliance with increasingly larger set of functional requirements at each stage of integration	Table 4
(b) assessment of timing, sizing, and accuracy	Table 4
(c) performance at boundaries and under stress conditions	Table 4
(d) measures of functional test coverage and software reliability	Table 4
Plan tracing of requirements to test design, cases, procedures, and execution results	Sec 6.4.6 (2)
Review plan documentation of test tasks and results	Sec 6.4.6 (2)
Input: SRS, SDD, Interface Requirements Documentation, Interface Design Documentation	Table 4
Output: Integration Test Plan, Anomaly Reports	Table 4
(5) Test Design Generation – SVVP Test Design Evaluation – deviations noted	Sec 6.4.7
Design tests for:	N/A
(a) component testing	Sec 6.4.7 (1-a)
Units	Table 4
Modules	Table 4
Sub-elements	Table 4
(b) integration testing	Sec 6.4.7 (1-b)
Sub-elements with sub-elements	Table 4
Hardware interfaces	Table 4
External software interfaces	Table 4
Functional requirements	Table 4
(c) system testing	Sec 6.4.7 (1-c)
End item functional requirements	Table 4
System requirements	Table 4
(d) acceptance testing	Sec 6.4.7 (1-d)
Software against acceptance criteria	Table 4

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
Continue tracing required by the Test Plan	Sec 6.4.7 (2)
Input: SDD, Interface Design Documentation, User Documentation	Sec 6.3, Table 4
Output: Component Test Design(s), Integration Test Design(s), System Test Design(s), Acceptance Test Design(s), Anomaly Reports	Sec 7.2, Table 4
Evaluate design safety analysis for correctness, consistency, completeness, accuracy, readability, and testability	Sec 6.4.1
Review plan documentation of analysis tasks and results	Sec 6.4.2
Input: Safety Analysis, SRS, SDD, Interface Requirements Documentation	Sec 6.4, Table 4
Output: Task Reporting, Anomaly Reports	Sec 7.2, Table 4
3.5.5 Implementation Phase V&V	Sec 6.5
(1) Source Code Traceability Analysis	Sec 6.5.2
Trace source code to corresponding design specification(s) and design specification(s) to source code	Sec 6.5.2 (1)
Analyze identified relationships for correctness, consistency, completeness, and accuracy	Sec 6.5.2 (2)
Input: SDD, Interface Design Documentation, Source Code Listing(s)	Table 4
Output: Task Reporting, Anomaly Reports	Table 4
(2) Source Code Evaluation	Sec 6.5.3
Evaluate source code for correctness, consistency, completeness, accuracy, and testability	Sec 6.5.3 (1)
Evaluate source code for compliance with established standards, practices, and conventions	Sec 6.5.3 (2)
Assess source code quality	Sec 6.5.3 (3)
Input: Source Code Listing(s), Standards (standards, practices, conventions), SDD, Interface Design Documentation, User Documentation	Table 4
Output: Task Reporting, Anomaly Reports	Table 4
(3) Source Code Interface Analysis	Sec 6.5.4

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
Evaluate source code with hardware, operator, and software interface design documentation for correctness, consistency, completeness, and accuracy	Sec 6.5.4 (1)
Analyze data items at each interface	Sec 6.5.4 (2)
Input: Source Code Listing(s), User Documentation	Table 4
Output: Task Reporting, Anomaly Reports	Table 4
(4) Source Code Documentation Evaluation	Sec 6.5.5
Evaluate draft code-related documents with source code to ensure completeness, correctness, and consistency	Sec 6.5.5 (1)
Input: Source Code Listing(s), User Documentation	Table 4
Output: Task Reporting, Anomaly Reports	Table 4
(5) Test Case Generation – SVVP Test Case Evaluation – deviations noted	Sec 6.5.8
Develop test cases for:	Defined in the STP
(a) component testing	Defined in the STP
(b) integration testing	Defined in the STP
(c) system testing	Defined in the STP
(d) acceptance testing	Defined in the STP
Continue tracing required by the Test Plan	The VVT will verify the traceability for Quality Class Q Software.
Input: SDD, Interface Design Documentation, Source Code Listing(s)	Table 4
Output: Component Test Cases, Integration Test Cases, System Test Cases, Acceptance Test Cases, Anomaly Reports	Table 4
(6) Test Procedure Generation – SVVP Test Procedure Evaluation – deviations noted	Sec 6.5.7
Develop test procedures for:	N/A See Sec 6.5.7
(a) component testing	N/A
(b) integration testing	N/A
(c) system testing	N/A

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
Continue tracing required by the Test Plan	N/A - The VVT is not responsible for tracing, only verification of it
Input: SDD, Interface Design Documentation, Source Code Listing(s), User Documentation	Table 4
Output: Component Test Procedures, Integration Test Procedures, System Test Procedures, Anomaly Reports	Table 4
(7) Component Test Execution – SVVP Component Test Evaluation – deviations noted	Defined in the STP
Perform component testing as required by component test procedures	Defined in the STP
Analyze results to determine that software correctly implements design	Defined in the STP
Document and trace results as required by the Test Plan	Defined in the STP
Input: Source Code Listing(s), Executable Code, SDD, Interface Design Documentation	Table 4
Output: Task Reporting, Anomaly Reports	Table 4
Evaluate implementation safety analysis for correctness, consistency, completeness, accuracy, readability, and testability	Sec 6.5.1
Review plan documentation of analysis tasks and results	Sec 6.5.1
Input: Safety Analysis, SRS, SDD, Interface Requirements Documentation	Table 4
Output: Task Reporting, Anomaly Reports	Table 4
3.5.6 Test Phase V&V	Sec. 6.6
(1) Test Procedure Generation – SVVP Acceptance Test Procedure Evaluation – deviations noted	Sec 6.6
Develop test procedures for acceptance test	N/A See Sec 6.6
Continue tracing required by the Test Plan	N/A See Sec 6.6
Input: SDD, Interface Design Documentation, Source Code Listing(s), User Documentation	Table 4
Output: Acceptance Test Procedures, Anomaly Reports	Sec 7.2, Table 4
(2a) Integration Test Execution	Sec 6.6.3

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
Perform integration testing in accordance with test procedures	N/A see Sec 6.6
Analyze results to determine if software implements software requirements and design and that software components function correctly together	Sec 6.6.3
Document and trace results as required by the Test Plan	N/A see Sec 6.6 & Sec 6.6.3 (2)
Input: Source Code Listing(s), Executable Code	Table 4
Output: Task Reporting, Anomaly Reports	Sec 7.2, Table 4
(2b) System Test Execution	Sec 6.6.4
Perform system testing in accordance with test procedures	N/A See Sec 6.6
Analyze results to determine if software satisfies system objectives	Sec 6.6.1
Document and trace all testing results as required by the Test Plan	Sec 6.6.4 (2)
Input: Source Code Listing(s), Executable Code, User Documentation	Table 4
Output: Task Reporting, Anomaly Reports	Sec 7.2, Table 4
(2c) Acceptance Test Execution	Sec 6.6.2
Perform acceptance testing in accordance with test procedures under formal configuration control	N/A see Sec 6.6
Analyze results to determine if software satisfies acceptance criteria	Sec 6.6.2 (1)
Document and trace all testing results as required by the Test Plan	Sec 6.6.2 (2)
Input: Source Code Listing(s), Executable Code, User Documentation	Table 4
Output: Task Reporting, Anomaly Reports	Sec 7.2, Table 4
Evaluate design safety analysis for correctness, consistency, completeness, accuracy, readability, and testability	Sec 6.6.1
Review plan documentation of analysis tasks and results	Sec 6.6.1
Input: Safety Analysis, SRS, SDD, Interface Requirements Documentation	Table 4
Output: Task Reporting, Anomaly Reports	Table 4
Evaluate system build documentation for correctness, consistency, completeness, accuracy, and testability	Sec 6.6.3
Review plan documentation of analysis tasks and results	Sec 6.6.3

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
Input: System Build Documentation, SRS, SDS, Source Code Listing(s)	Table 4
Output: Task Reporting, Anomaly Reports	Sec 7.2, Table 4
3.5.7 Installation and Checkout Phase V&V	Sec 6.7
(1) Installation Configuration Audit	Sec 6.7.1
Audit installation package to determine that all software products required to correctly install and operate the software are present	Sec 6.7.1 (1)
Analyze all site-dependent parameters or conditions to determine that supplied values are correct	Sec 6.7.1 (2)
Conduct analyses or tests to demonstrate that installed software corresponds to software subjected to V&V	Sec 6.7.1 (3)
Input: Installation Package	Table 4
Output: Task Reporting, Anomaly Reports	Sec 7.2, Table 4
(2) V&V Final Report Generation	Sec 7.4
Summarize all V&V activities and results, including status and disposition of anomalies in the V&V final report	Sec 7.4
Input: All V&V Phase Summary Reports	Table 4
Output V&V Final Report	Table 4
Evaluate installation and checkout safety analysis for correctness, consistency, completeness, accuracy, readability, and testability	Sec 6.7.2(1)
Review plan documentation of analysis tasks and results	Sec 6.7.2
Input: Safety Analysis, SRS, SDD, Interface Requirements Documentation	Table 4
Output: Task Reporting, Anomaly Reports	Sec 7.2, Table 4
3.5.8. Operation and Maintenance Phase V&V	N/A to this plan. The SMP and this plan do not consider the Operation and Maintenance Phases.
3.6 Software Verification and Validation	N/A
3.6.1. Required Reports	Sec 7
(1) Task Reporting	Sec 7.1
(2) V&V Phase summary Report	Sec 7.3

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
(a) Description of V&V tasks performed	Sec 7.3 (2)
(b) Summary of task results	Sec 7.3 (2)
(c) Summary of anomalies	Sec 7.3 (1)
(d) Recommendations	Sec 7.3 (3)
(3) Anomaly Reporting	Sec 7.2
(a) Description & location	Sec 7.2 (6c)
(b) Impact	Sec 7.2 (6c)
(c) Criticality	Sec 7.2 (6b)
(d) Recommendations	Sec 7.2 (6c)
(4) V&V Final Report	Sec 7.4
(a) Summary of all life cycle V&V tasks	Sec 7.4 (1)
(b) Summary of task results	Sec 7.4 (2)
(c) Summary of anomalies and resolutions	Sec 7.4 (3)
(d) Assessment of overall software quality	Sec 7.4 (4)
(e) Recommendations	Sec 7.4 (5)
3.6.2 Optional Reports	N/A Optional reports are not to be considered by the SVVP
(1) Special Studies reports	N/A
(a) Purpose and Objectives	N/A
(b) Approach	N/A
(c) Summary and results	N/A
(2) Other reports	N/A
3.7 Verification and Validation Administration	Sec 8
3.7.1 Anomaly Reporting and Resolution	Sec 8.2
3.7.2 Task Iteration Policy	Sec 8.4
3.7.3 Deviation Policy	Sec 8.5
3.7.4 Control Procedures	Sec 8.6

Table 3 SVVP Cross Reference to IEEE 1012 [2.3.2(1)] (Continued)

IEEE 1012 Requirement	SVVP Section, Paragraph
3.7.5 Standards, Practices, and Conventions	Sec 8.7

Table 4 Life Cycle Phase Output Assessment Requirements

Task	Method	Acceptance Criteria*	Inputs	Outputs
PLANNING				
Software Management Plan	Independent Verification	Conformance with BTP-14 (3.1.a), NR-6101 (4.1.1)	MMIS Design Implementation Plan	Design Verification Package
Software QA Plan	Independent Verification	Conformance with BTP-14 (3.1.c), NR-6101 (4.1.2)	Quality Assurance Program, MMIS Design Implementation Plan	Design Verification Package
Software Configuration Management Plan	Independent Verification	Conformance with BTP-14 (3,1,k), NR-6101 (4.1.3)	MMIS Design Implementation Plan	Design Verification Package
Software Verification and Validation Plan	Independent Verification	Conformance with BTP-14 (3.1.j), NR-6101 (4.1.4)	MMIS Design Implementation Plan	Design Verification Package
Software Safety Analysis Plan	Independent Verification	Conformance with BTP-14 (3.1.i), NR-6101 (4.1.5)	DCIS Software Safety Plan	Design Verification Package
Software Development Project Plan	Independent Verification	Conformance with BTP-14 (3.1.b), NR-6101 (4.1.6.3)	Software Development Project Plan	Design Verification Package
Software Integration Plan	Independent Verification	Conformance with BTP-14 (3.1.d), NR-6101 (4.1.7.3)	Software Integration Plan	Design Verification Package
Software Installation Plan	Independent Verification	Conformance with BTP-14 (3.1.e), NR-6101 (4.1.8.3)	Software Installation Plan	Design Verification Package
Planning Phase Safety Analysis	Independent Verification	Conformance with BTP-14 (3.2.a), Software Safety Analysis Plan	Software Safety Analysis Plan, Software Development Project Plan, System Definition Documents	Design Verification Package

Table 4 Life Cycle Phase Output Assessment Requirements (Continued)

Task	Method	Acceptance Criteria*	Inputs	Outputs
REQUIREMENTS				
Software Requirements Specification	Independent Verification	Conformance with NR-6101 (4.2.1), SMP	Software Requirements Specification	Design Verification Package
Requirements Safety Analysis	Independent Verification	Conformance with Software Safety Conformance with Analysis Plan, NR-6101 (4.2.2)	Requirements Safety Analysis Plan, Safety Analysis Report	Design Verification Package
Hardware/Software Specification	Independent Verification	Conformance with SMP & NR-6101 (4.2.1) when no SRS exists. SDD, LDs and I/O Lists, if applicable, P&IDs	Hardware/Software Specification	Design Verification Package
System Block Diagram	Independent Verification	SDD and LDs, if applicable, P&IDs	System Block Diagram	Design Verification Package
Inst. Performance Specification	Independent Verification	HSS, SBD	Instr Performance Specification	Design Verification Package
Sub-System Schematic	Independent Verification	HSS, if applicable, LDs, P&IDs, and IPS	Sub-System Schematic	Design Verification Package
Ext Data Comm Protocol Spec	Independent Verification	HSS or SRS	Ext Data Comm Protocol Spec	Design Verification Package
Software Validation Test Plan	Independent Verification	Conformance with SMP	Software Validation Test Plan	Design Verification Package
User's Manual	Independent Verification	HSS, if applicable, SBD and IPS	SRS, Interface Requirements Documentation, User Documentation	Design Verification Package
Support S/W, Tool Documentation	Audit Review of Operating Experience Independent Verification	Performed according to the instructions in SMP, SQAP SCMP, SVVP and SSP	Support S/W, Tool Documentation	Design Verification Package

Table 4 Life Cycle Phase Output Assessment Requirements (Continued)

Task	Method	Acceptance Criteria*	Inputs	Outputs
Third Party S/W Documentation	Audit Review of Operating Experience Independent Verification	Performed according to the instructions in SMP, SQAP SCMP, SVVP and SSP	Third Party S/W Documentation	Design Verification Package
Prev Dev S/W Documentation	Audit Review of Operating Experience Independent Verification	Performed according to the instructions in SMP, SQAP SCMP, SVVP and SSP	Prev Dev S/W Documentation	Design Verification Package Anomaly Report
DESIGN				
Software Architecture Specification	Independent Verification	User's Manuals, External Data Communication Protocol Specification, and either SRS (software-based system), or SBD, Sub-system Schematic and software section of IPS (microprocessor-based system)	Software Architecture Specification	Design Verification Package
Software Design Specification	Independent Verification	User's Manuals, External Data Communication Protocol Specification, and either SRS (software-based system), or SBD, Sub-system Schematic and software section of IPS (microprocessor-based system)	Software Design Specification	Design Verification Package
Design Safety Analysis	Independent Verification	Conformance with Software Safety Analysis Plan, NR-6101 (4.3.3)	SDS & Architecture Specification	Design Verification Package
Internal Data Comm Protocol Specification	Independent Verification	IPS	Internal Data Comm Protocol Specification	Design Verification Package
Validation Test Procedures and Test Cases	Independent Verification	HSS, User's Manual and Software Test Plan, for software-based systems,	Validation Test Procedures and Test Cases	Design Verification Package

Table 4 Life Cycle Phase Output Assessment Requirements (Continued)

Task	Method	Acceptance Criteria*	Inputs	Outputs
		SRSs, for microprocessor-based systems, SBD and IPSs.		
Software Conventions and Guidelines	Independent Verification	Performed according to the instructions in SMP, SQAP SCMP, SVVP and SSP	Software Conventions and Guidelines	Design Verification Package
If applicable, Support Software/Tool documentation package	Independent Verification	Performed according to the instructions in SMP, SQAP SCMP, SVVP and SSP	Support Software/Tool documentation	Design Verification Package
If applicable, Third Party Software documentation package	Independent Verification	Performed according to the instructions in SMP, SQAP SCMP, SVVP and SSP	Third Party Software documentation	Design Verification Package
If applicable, supplemental documentation for Previously Developed Software	Independent Verification	Performed according to the instructions in SMP, SQAP SCMP, SVVP and SSP	Documentation for Previously Developed Software	Design Verification Package
IMPLEMENTATION				
Code Listings	Inspection Independent Verification	In accordance with the SQAP and the SDS, Software Conventions and Guidelines document, and for microprocessor-based system, Internal Data Communication Protocol Specification	Computer Code Listings	Design Verification Package
Code Safety Analysis	Independent Verification	Conformance with Software Safety Analysis Plan, NR-6101 (4.4.2)	Software Safety Analysis Plan, Safety Analysis Report	Design Verification Package
Software Module Testing	Concurrent and /or supplemental Testing	Conformance with SMP	Software Module Test Plan and Procedure	Design Verification Package
Module Test Report	Independent Verification	SDS, Software Test Plan	Module Test Reports	Design Verification Package

Table 4 Life Cycle Phase Output Assessment Requirements (Continued)

Task	Method	Acceptance Criteria*	Inputs	Outputs
INTEGRATION TEST				
Integration Safety Analysis	Independent Verification	Conformance with Software Safety Analysis Plan, NR-6101 (4.5.2)	Integration and Installation Test Report	Design Verification Package
Integration Testing	Testing	Conformance with SMP	Integration Testing Plan and Procedure	Design Verification Package
Integration and Installation Test Report	Independent Verification	Conformance with SMP	Integration and Installation Test Report	Design Verification Package
Acceptance Test Procedure	Audit Independent Verification	Conformance with Section 6.6.2	Acceptance Test Plan and Procedure	Design Verification Package
Acceptance Test Results	Independent Verification	Conformance with Acceptance Test Procedure	Acceptance Test Results	Design Verification Package
VALIDATION TEST				
Validation Safety Analysis	Independent Verification	Conformance with Software Safety Analysis Plan, NR-6101 (4.6.1)	Validation Test report	Design Verification Package
Validation Testing	Testing	Test Procedure	Validation Test Plan and Procedure	Validation Test Report
Validation Test Report	Independent Verification	HSS, Software Test Plan, for software-based system, SRSs, for microprocessor-based system, SBD and IPS	Validation Test Report	Design Verification Package
If applicable, Support Software/Tool documentation package	Independent Verification	Performed according to the instructions in SMP, SQAP, SCMP, SVVP and SSP	Support Software/Tool documentation package	Design Verification Package
Build Release Description	Independent Verification	Performed according to the instructions in SMP, SQAP, SCMP, SVVP and SSP	Build Release Description	Design Verification Package

References: NR-6101 [2.3.3(2)]
BTP-14 [2.3.3(3)]

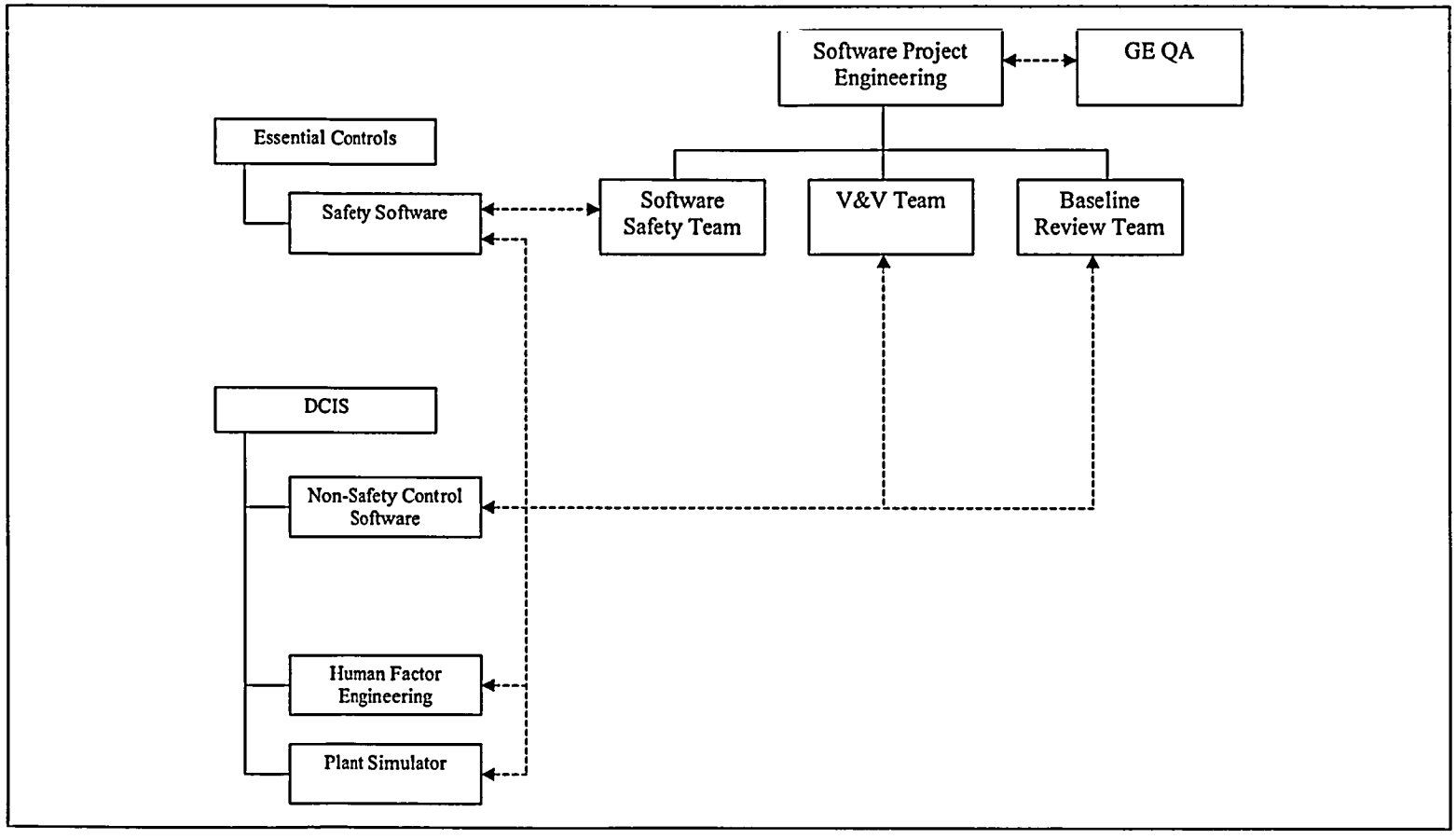

Figure 1 Design Organizations and SPE Interface Chart