GE Energy
Nuclear

NEDO-33230
Class I
eDRF# 0000-0050-2665
January 2006

LICENSING TOPICAL REPORT

ESBWR I&C
SOFTWARE SAFETY PLAN

INFORMATION NOTICE

This document NEDO-33230, Rev. 0, 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

Table of Contents

List of Tables

1 Introduction

1.1 Purpose and Scope

This Software Safety Plan (SSP) establishes the processes and activities intended to ensure the safety of the Safety-related Software for the ESBWR project in coordination with the supporting documentation listed in Section 2 of this plan. Safety is the most important consideration for the project, taking precedence over budget and schedule.

The software-based products covered in this plan encompass all instrumentation and control systems which perform the monitoring, control and protection functions associated with all modes of ESBWR plant normal operation (i.e., startup, shutdown, standby, power operation, and refueling) as well as off-normal, emergency and accident conditions.

Assessing the adequacy of core inputs and logic defined outside of the software scope is outside the scope of the Software Safety Analysis. Thus, as examples, the initiation logic for the SLC system and the adequacy of the SLC system itself to perform the core cooling function is beyond the scope of the Software Safety Analysis. This is because as defined it may be implemented in software or in hardware. However, logic, architecture, and design details of applying such inputs to a software implementation of such logic is with-in the scope of the Software Safety Analysis.

1.2 Safety Goals

The purpose of the SSA is to ensure compliance with safety goals established for the ESBWR Nuclear Power Plant. The principal safety goals associated with the instrumentation and control systems for the ESBWR Plant are identified in the DCD.

2 Definitions, Acronyms and Abbreviations, and References

2.1 Definitions

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.
Failure Mode and Effects Analysis	A tabular method of providing traceability from the modes by which a system may fail and the effect of that failure on the ability of the system to perform its function, or the ability of a collection of systems to recover from the failure.
Fault Tree	A pictorial method of providing traceability from the modes by which a system may fail and the effect of that failure on the ability of the system to perform its function, or the ability of a collection of systems to recover from the failure.

Hazardous State	Any degraded system state, which affects the ability of the system to perform a function important to safety. A hazardous state may or may not be detectable before the system is called upon to perform the function.
Regression Testing	Regression testing demonstrates that changes to the software did not introduce conditions for new hazards. Regression testing for the Project is to be performed per the requirements established in SVVP.
Safety Critical Software	Safety critical software falls into one or more of the following categories: 1. Software whose inadvertent response to stimuli, failure to respond when required, response out-of-sequence, or response in combination with other responses can result in an accident. 2. Software that is intended to mitigate the result of an accident. 3. Software that is intended to recover from the result of an accident.
Software Life Cycle	The period of time that begins when a software product is conceived and ends when the software is no longer available for use (See Section 3.4).
Software Package	A collection of modules (e.g., subroutines, main control tasks) brought together to form a single software product.
Stress Testing	Stress testing demonstrates that the software will not cause hazards under abnormal circumstances, such as unexpected input values or overload conditions. Stress testing for the Project will be performed as part of both Integration Testing and Validation Testing per the requirements established in the SMP.
Traceability Matrix	A matrix that records the relationships between two or more products of the development process (e.g. a matrix that records the relationship between the requirements and the design of a given software component).

2.2 Acronyms

ATWS	Anticipated Transients without Scram
DCD	Design Control Document
DRF	Design Record File
FMEA	Failure Modes and Effects Analysis
GE	General Electric Company
HSS	Hardware/Software Specification
I&C	Instrumentation and Control
I/O	Input/Output
IEEE	Institute of Electrical and Electronic Engineers
IFAT	Integrated Factory Acceptance Test
IPS	Instrument Performance Specification
MMIS	Man Machine Interface System
PDM	Project Design Manual
PHA	Preliminary Hazards Analysis
PRA	Probabilistic Risk Assessment
RSSE	Responsible Software Safety Engineer
SBD	System Block Diagram
SCMP	Software Configuration Management Plan
SDD	System Design Description
SDP	Software Development Plan
SDS	Software Design Specification
SFRA	System Functional Requirements Analysis
SLC	Standby Liquid Control

SMP	Software Management Plan
SRS	Software Requirements Specification
SSA	Software Safety Analysis
SSP	Software Safety Plan
SVVP	Software Verification and Validation Plan
VVT	Verification and Validation Team

2.3 References

2.3.1 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. Design Control Document (DCD)	A11-5299
2. Project Design Manual (PDM)	
3. Software Management Plan (SMP)	NEDO-33226
4. Man-Machine Interface System and HFE Design Implementation Plan (MMIS/HFE IP)	NEDO-33217
5. NP 2010 COL Demonstration Project Quality Assurance Program	NEDO-33181
6. Composite Specification (26A6007)	A11-5299

2.3.2 Supplemental Documents

The following supplemental documents are used in conjunction with this document.

Document Title	Document Number
1. ESBWR Probabilistic Risk Assessment (PRA)	
2. Software Development Plan (SDP)	NEDO-33229
3. Software Verification and Validation Plan (SVVP)	NEDO-33228

Document Title	Document Number
4. Software Configuration Management Plan (SCMP)	NEDO-33227
5. GE Nuclear Energy Engineering Operating Procedures	NEDE-21109
a. 40-7.00 Design Review	
b. 42-6.00 Independent Design Verification	
c. 42-10.00 Design Record File	
d. 70-42 Reporting of Defects and Noncompliance Under 10CFR Part 21	
6. Software Quality Assurance Plan (SQAP)	NEDO-33245

2.3.3 Codes and Standards

The following codes and standards are applicable to the Software Safety Plan to the extent specified herein. The applicable date/revision of the code or standard is specified in the Composite Specification [2.3.1 (6)].

Institute of Electrical and Electronic Engineers (IEEE) Standards

1. IEEE 1228, Software Safety Plans

3 Software Safety Management

This section of the Plan describes the organization, schedule, resources, responsibilities, tools, techniques, and methodologies used in the safety assessment for safety-critical software.

3.1 Organization and Responsibilities

The Software Safety Team is part of the Software Project Engineering Organization as described in the Software Quality Assurance Plan (SQAP). The Software Management Plan (SMP) provides a description of the organization, including the software safety organization. Software safety activities are performed in parallel to the design process.

The Responsible Software Safety Engineer (RSSE) has responsibility for assuring the safety qualities of the software being developed, including the interface between the hardware and software. The RSSE has the authority to enforce safety requirements in the software requirements specification, the design, and the implementation of the software. The RSSE also has the authority to reject the use of pre-developed software if the software can be shown to be unsafe or a tool, if it can be shown that the tool will adversely impact the safety of the final software system.

The RSSE will verify all SSA activities documented by the design team to ensure that safety is adequately addressed at all stages in the development process. In particular, the RSSE shall facilitate the interchange of observations between safety analyses for different systems and ensure that any hazards impacting or identified by a subcontractor are communicated to the affected organizations. The RSSE will also monitor the performance of the analyses by reviewing comments made during the verification of the analyses and providing supplemental training to engineers performing the SSA as indicated by any observed trends in those comments. The RSSE will also monitor the progress of the analysis, obtaining additional resources as needed, in order to maintain the overall project schedule

3.2 Resources

The Software Development Plan (referred to as SDP) specifies management process for the design and delivery activities for the safety software and hardware. The Responsible Software Safety Engineer (RSSE) shall determine the resource requirements for the implementation of the software safety analysis in accordance with the process specified in the SDP at the beginning of each phase of development. Resource Requirements shall be documented in the DRF.

3.3 Staff Qualification and Training

The SQAP defines the staff qualification requirements. The RSSE will ensure that all engineers performing software safety analysis are familiar with the Software Safety Plan (this document) and will conduct team meetings, which will include discussion of any questions or developing concerns about the Software Safety Analysis.

3.4 Software Life Cycle

The software life cycle to be used in the development of the DCIS software is described in the SMP.

It is noted that the Software Life Cycle established for the project has broken testing into three separate groups. Module Testing is grouped with Code Evaluation, while Integration Testing and Validation Testing are performed as separate functions. This plan deviates slightly from the format recommended in IEEE-1228 in that the requirements for the SSA of these testing activities is addressed according to the Software Life Cycle, rather than grouping all testing requirements into one activity.

3.5 Documentation Requirements

A summary of the documentation to be prepared for the software safety analysis is supplied below.

1. Software Project Management - Documentation of how the Software Safety Program will be implemented, integrated and managed with other development activities is discussed in this document (SSP) and in the SMP.

2. Software Configuration Management - Information regarding the configuration management of software safety-related modules and documents is provided in the SCMP.

3. Software Quality Assurance - Information related to the quality assurance requirements are primarily located in the SMP, with supplemental information contained in the SCMP, SVVP, SQAP and SDP. Section 3.5 of this plan provides information regarding the location of the various elements of software quality assurance within the planning documents.

4. Software Safety Requirements - The safety requirements will be established in the Hardware/Software Specifications and the Software Requirements Specifications or Instrument Performance Specifications in accordance with the requirements established in the SMP.

5. Software Safety Design - The software design activities will be documented in the Software Design Specifications in accordance with the SMP.

6. Software Development Methodology, Standards, Practices, Metrics and Conventions - The practices that are to be used in the safety-related software development will be documented in the Software Conventions and Guidelines Document in accordance with the requirements established in the SMP.

7. Test Documentation - Software safety-related test planning, test design, test procedures and test reports will be developed as documented in the SMP.

8. Software Verification and Validation - Traceability of safety requirements throughout the software development process will be provided in accordance with Section 4 of this plan. Information regarding how software will be verified and validated is provided in the Software Verification and Validation Plan (SVVP).

9. Reporting Safety Verification and Validation - The reporting requirements for documenting the results of the software safety-related verification and validation activities are provided in the SVVP.

10. Software User Documentation - User's Manuals will be prepared as specified in the SMP.

11. Verification of Software Safety Requirements Analysis - Requirements for this activity are specified in Section 4.2 of this SSP.

12. Verification of Software Safety Design Analysis - Requirements for this activity, which will include module testing are specified in Section 4.3 of this SSP.

13. Verification of Software Safety Code Evaluation - Requirements for this activity are specified in Section 4.4 of this SSP.

14. Verification of Software Safety Test Analysis - Requirements for this activity are specified in Section 4.5 and 4.6 of this SSP.

15. Verification of Software Safety Change Analysis - Requirements for this activity are specified in Section 4.7 of this SSP.

The Software Configuration Management Plan specifies the process for control of updates to all documents specified, including any revisions to this plan itself.

3.6 Software Safety Program Records

Software safety analysis shall be documented by the design team in phase appropriate documents (as described in section 3.5). The minimum software safety program records to be generated, maintained and retained by the project, as well as the responsibility for these activities, are documented in SCMP.

Tracking of software hazards will be performed via the Software Safety Analysis in these documents.

3.7 Software Configuration Management Activities

All safety-critical software will be configuration managed in accordance with the Software Configuration Management Plan (SCMP). This plan applies to all activities to be implemented during the development of safety-related software-based products.

Although documentation of the development activities will be maintained after turnover in accordance with the requirements of the SCMP, configuration management activities after the turnover of the plant is beyond the scope of the DCIS supplier and shall be established separately.

3.8 Software Quality Assurance Activities

This plan, in coordination with the SMP, SQAP, SCMP, and SVVP describe the quality assurance activities for safety software employed in the project.

3.9 Software Verification and Validation Activities

The SMP, describes the engineering process, which will be used in the development of the software. This process provides a methodical set of life cycle activities designed to ensure that the software will operate as required. Verification and Validation will be performed in accordance with the SVVP. The SST verification of SSA life cycle

activities will verify the system safety requirements and system hazard analysis to ensure that:

1. All system safety requirements have been satisfied by the life cycle phases, and
2. No additional hazards have been introduced by the work done during the life cycle activity.

3.10 Tool Support and Approval

All tools used in the development and evaluation of safety-related software shall be evaluated for suitability. Any Support Software used to "support" the development or evaluation of the software shall be managed as specified in the SMP. Configuration control of Support Software shall be managed per the requirements of the SCMP.

3.11 Previously Developed or Purchased Software

Any acquired software shall be managed per the requirements of the SMP, SCMP and SVVP.

3.12 Subcontract Management

Management of subcontractors for Safety-Related Software shall be carried out in accordance with the requirements of the SVVP. Any software safety activities performed by a subcontractor will either be performed in accordance with this plan or an alternate plan approved by GE before the performance of the activity.

The RSSE shall be responsible for ensuring that hazards impacting or identified by the subcontractor are communicated to any affected organizations.

3.13 Process Certification

The process certification activity specified in IEEE-1228 is equivalent to the Baseline Reviews performed for the project. Baseline Review Team will document the Software Safety Activities as complying with the requirements of the processes here and in the SMP, SCMP, SVVP, and SQAP.

4 Software Safety Analyses

This section identifies the requirements for Software Safety Analyses to be performed as a part of the software development process.

4.1 Planning Phase Analysis

4.1.1 Task Description

The purpose of the Software Safety Analysis (SSA) of the Planning Phase is to establish the fundamental Nuclear Power Plant design characteristics as they affect the design and implementation of software used in plant systems designated as safety-related or 'Quality Class Q'. This phase of development is referred to as analysis preparation in IEEE-1228.

Generically, this is the identification of all software systems and classification of which should be designated as safety-related or non-safety-related.

Since this has been done regardless of the use of software, the Software Safety Team can use these as the analysis preparation or PHA. At this phase, safety systems may or may not be implemented using software. Safety systems that are identified as having no software in their implementation by the design organization shall become outside the scope of this plan.

The SST shall review the ESBWR DCD, PRA, SFRA to confirm that for safety systems it identifies:

a. Hazardous plant states

b. Sequences of actions that can cause the plant to enter a hazardous state

c. Sequences of actions intended to return the system from a hazardous state to a non-hazardous state

d. Actions intended to mitigate the consequences of accidents

2. An evaluation of the high-level system design identifying those functions that will be performed by software and specifying the software safety-related and safety critical actions that will be required of the software to prevent the system from entering a hazardous state, or to move the system from a hazardous state to a non-hazardous state, or to mitigate the consequences of an accident.

3. The interfaces between the software and the rest of the system

4. An evaluation of the adequacy of the System Design Descriptions and Logic diagrams for all Essential Controls Systems.

4.1.2 Input Documents

The inputs to SST planning phase are as follows

1. PDM

2. Design Control Document (DCD)

3. Probabilistic Risk Assessment (PRA)

4. System Functional Requirements Analysis (SFRA)

5. System Design Specifications (SDS) or System Design Descriptions (SDD)

6. Logic Diagrams (LD)

4.1.3 Outputs

The review described in section 4.1.1 shall document;

1. A list of high level system safety functions that will be performed by software.
2. A list of all interfaces between the software and the rest of the system
3. Documentation of any findings regarding specification documents defined in Subsection 4.1.2.2

4.1.4 Methods

The documentation of the review shall be filed in the Design Record File.

4.1.4.1 Identification of Systems

All plant systems will be reviewed to determine which systems are considered safety-related (Quality Class Q). Systems that meet any of these criteria will be included in the list of systems for further evaluation. If the system has been identified as employing software it should be indicated. This table should be periodically reviewed with the design team to identify any changes of the use of software in safety systems. A table will be prepared containing the information shown here.

System Number	System Name	Quality Class	Direct Initiator?	Uses Software
X00	Name	N/Q	Yes/No	Yes/No

4.1.4.2 Identify System Functions

The next step in the process is to identify all of the high-level safety functions performed by software identified from the previous step (4.1.4.1). Not all software functions of these systems are Quality Class Q, but they shall be included to assure that implementation does not introduce new safety issues. The System Functional Requirements Analysis Reports (SFRAs) developed for the Human Factors Evaluation documents an extensive review of the requirements and design documents to identify all System Functions. Therefore, a review of the SFRAs will be performed to help identify software implemented in safety systems.

The output of this step is a table containing the following information.

Function Number	Function	DCIS System Name	Interfacing System	Hardware or Software
DCIS System Number - sequential number	Function name	System Name	System Name	H/S

4.1.4.3 Identify Software Interfaces

The next step in the planning phase focuses on the interface between the software and the rest of the system.

4.1.4.4 Develop PHA

For the ESBWR project the DCD and PRA contain the required information for a PHA and as such they shall be used for the PHA.

4.1.5 Reporting Requirements

Reporting shall be done in accordance with verification reporting requirements specified in the SVVP and EOP 42-6.0. Planning activities shall be documented and stored in the project DRF in accordance with project planning EOPs.

4.2 Requirements Phase Analysis

4.2.1 Task Description

This section defines the software safety-related activities that will be carried out as part of the Requirements Phase of development. This phase of development is referred to as requirements analysis in IEEE-1228.

The requirement phase tasks, which occur in this phase of the analysis, are established in the SMP. These tasks will establish the design constraints and guidelines, such as system algorithms, response time characteristics, and sizing. Validation test requirements will also be established in this life cycle phase. These documents shall contain the software safety analysis for all Quality Class Q systems, regardless of the software classification. The SSA section of these documents will be verified by the SST as described in this plan and the SVVP.

4.2.1.1 Specification Analysis

Specification analysis evaluates each safety-critical software requirement has been analyzed against key safety qualities. The SST shall have a scope of verification on all software safety specifications with respect to the verification of qualities, such as, completeness, correctness, consistency, testability, robustness, integrity, reliability,

usability, flexibility, maintainability, portability, interoperability, accuracy, audit ability, performance, internal instrumentation, security and training. Specification shall demonstrate consideration of these qualities as well as documenting an analysis of safety system hazards such as abnormal operating conditions.

4.2.1.2 Timing and Sizing Analysis

Timing and sizing analysis evaluates safety implications of safety-critical requirements that relate to execution time, clock time, and memory allocation. These evaluations are performed in the preparation of the HSS/SRS/IPS and related documents. The SST shall have a scope of verification of any document containing timing and sizing analysis to verify that that all such analysis is incorporated in the documents hazard analysis.

4.2.1.3 Criticality Analysis

Criticality analysis identifies all software requirements that have safety implications. Each requirement in the Hardware/Software Specification (HSS), Software Requirements Specification (SRS) and/or Instrument Performance Specification (IPS) shall have a safety classification and shall be verified against the PHA documentation to assess its potential for unacceptable risk.

4.2.1.4 External Interface Analysis

Interface analysis ensures the proper design of a software component's safety-critical interfaces with other components of the system, both internal and external. For the project, the definition of the external interfaces is performed during the requirement phase. Therefore, the External Interface Analysis is performed in the requirements phase and documented in the appropriate requirements phase specification (HSS, SRS/IPS and/or External Data Communications Protocol Specifications). The Internal Interface Analysis will be performed during the Design Phase described in Section 4.3.

The major areas of concern with interfaces are properly defined protocols and control and data linkages. Hazards associated with an interface are also related to the system context and the environmental context as defined by their state at any point in time. The interface analysis will be verified by the SST to ensure that these system and environment contexts are documented.

4.2.1.5 Different Software System Analysis

IEEE-1228 indicates that different software system analysis may be required if more than one software system is being integrated, particularly for conditions when a system hazard results in a requirement that is partially implemented in two or more software systems. Defense-in-Depth and Diversity of plant functions is the basis for the plant design, and as such, overlapping system function should not be an issue. The remaining concern about multiple systems is the careful identification of the interfaces between different software

systems, which is evaluated in the External Interface Analysis described in Subsection 4.2.1.4. Therefore, for the plant, no analysis of different software systems is required.

4.2.1.6 Hazards Analysis

A hazards evaluation will be performed, extending the preliminary hazards analysis (using inputs from the PRA & DCD). Failure Mode and Effects Analysis (FMEA) methods and/or Fault Tree methods may be used. This evaluation will use the insights obtained from the Specification Analysis, Timing and Sizing Analysis, Criticality Analysis and External Interface Analysis to provide additional detail in the hazards evaluation. A hazards analysis shall be provided in the HSS and SRS/IPS and shall be verified by the SST to confirm that safety-related aspect of the identified in the PHA are considered. Additionally, each requirement in the HSS and SRS/IPS will be examined to determine if any additional hazards are introduced into the design. If this verification identifies a new failure mode of the system, then the planning documentation will be revised to incorporate this failure mode.

4.2.1.7 Reliability Goals

The criticality level of each safety-related requirement should be examined based on the Hazards Analysis. This information may then be factored into reliability and availability assessments.

4.2.1.8 Test Plan Analysis

The SST shall have a scope of verification of the Software Integration Plan to ensure that the plan provides adequate guidance for software development team with respect to testing safety software. The plan should address testing of safety-critical software, and include third party software as well as previously developed software.

4.2.1.9 Security Analysis

Security requires that security threats to the computer system be identified and classified according to severity and likelihood. Actions required of the software to detect, prevent, or mitigate such security threats should be specified, including access control restrictions. The SST shall have a scope of verification of the SRS/IPS to confirm that security threats to safety-related software systems have been considered and documented.

4.2.2 Input Documents

Inputs to SST requirement phase activities are defined as follows:

1. Hardware/Software Specifications (HSS)
2. Software Requirements (SRS)
3. System Block Diagram (SBD)
4. Instrument Performance Specifications (IPS)

5. Subsystem Schematics
6. External Data Communication Protocol Specification(s)
7. Software Integration Plan (SIntP) or Software Test Plan
8. User's Manuals

4.2.3 Outputs

All SST activities in this phase cumulate with verification activities. Results of all verification activities shall be documented in the verification package.

4.2.4 Methods

The Specification Analysis, Timing and Sizing Analysis, and the Criticality Analysis, and the External Interface Analysis will be performed and documented by the design organization as specified in this plan. The hazards evaluation should be performed using FMEA and/or fault tree techniques. Verification methods are described in the SVVP.

4.2.5 Reporting Requirements

Reporting shall be done in accordance with verification reporting requirements specified in the SVVP and EOP 42-6.0.

4.3 Software Design Analysis

4.3.1 Task Description

The software safety analysis of the software design ensures that the safety-critical portion of the software design correctly implements the safety-critical requirements and introduces no new hazards. This phase of development is referred to as design analysis in IEEE-1228.

The purpose of the software design is to provide adequate information for an experienced software engineer to write the software program. For previously developed software, all required modifications would be defined. The validation test procedures and test case specification will also be developed during this phase of the analysis.

Separate analyses will be performed for each software system. The SST shall verify analysis activities are documented in the SDS and other documents from this life cycle phase. Analyses may include, but are not limited to, those listed below.

4.3.1.1 Functional Analysis

Functional analysis ensures that each safety-critical software requirement is covered and that an appropriate criticality level is assigned to each software element. This analysis shall be performed by the SST as part of the verification scope of the SDS.

4.3.1.2 Logic Analysis

The logic (which identifies the safety-critical equations, algorithms, and control logic of the software design) is documented in the SDSs. The SDSs also describe the partitioning of the functional requirements of the system into tasks. Logic Analysis establishes traceability from these tasks back to the requirements established requirements phase, evaluates the ability of the architecture to provide the redundancy and separation goals, and determines if any additional hazards are introduced. The SST shall have a scope in the verification of the SDS to confirm the traceability of logic requirements and that no new hazards are introduced.

4.3.1.3 Data Analysis

Data analysis evaluates the description and intended use of each data item in the software design. This analysis ensures that the structure and intended use of data will not result in a hazard. The SST shall have a scope of verification of the SDS to confirm that:

1. data dependencies in data structures that that circumvent isolation, partitioning, data aliasing, and fault containment issues affecting safety, and

2. the control or mitigation of hazards have been defined.

4.3.1.4 Internal Interface Analysis

Interface analysis verifies the proper design of a software component's safety-critical interfaces with other components of the system, both internal and external. External interfaces will be verified during the Requirement Phase, as documented in Section 4.2 of this plan. The SST shall have a scope of verification of the SDS and/or the Internal Protocol Specification to confirm that:

1. protocols are properly defined for interfaces between software within a given system to ensure that and that the control and data linkages are correct, and

2. no additional system-level hazards are introduced through the interfaces.

4.3.1.5 Constraint Analysis

Constraint analysis evaluates the safety of restrictions imposed on the selected design by the requirements and by real-world restrictions. The impacts of the environment on this analysis can include such items as the location and relation of clocks to circuit cards, the timing of a bus latch when using the longest safety-related timing to fetch data from the most remote circuit card, interrupts going unsatisfied due to a data flood at an input, and human reaction time. The impact of constraints is included in the SDS and will be considered in determining the possible failure modes of the software. The SST shall have a scope in the verification of the SDS to confirm the constraints have been defined and considered as specified herein.

4.3.1.6 Non-critical Code Analysis

Non-critical code analysis examines software elements that are not designated safety-critical and ensures that these elements do not cause a hazard. As a general rule, safety-critical code should be isolated from non-safety code. If isolation is not provable, a discussion of how to handle the risk should be provided. This analysis shall be documented in the SDS and the SST shall have a scope to verify the analysis.

4.3.1.7 Software Element Analysis

This analysis is performed as part of the non-critical code analysis.

4.3.1.8 Timing and Sizing Analysis

Timing and sizing analysis performed during the Requirement Analysis are further defined with respect the design and are documented in the SDS. The SST shall have a scope of verification of any document containing timing and sizing analysis to verify that that all such analysis is incorporated in the documents hazard analysis.

4.3.1.9 Test Procedure Evaluation

The Validation Test Procedures are developed in this phase of the software life cycle. The SST shall have a scope of verification for safety system validation test procedures to confirm that:

1. the procedure documents the ability to validate that the software will operate as intended.

2. any task, or subtask, which cannot be tested, is documented and the impact on safety has been determined.

4.3.1.10 Reliability and Availability Assessments

Reliability predictions may be made for availability- and reliability-critical software elements. The SST shall have a scope of verification of the SDS to confirm that assessments are considered.

4.3.1.11 Coding Practice Evaluation

In this phase of the software life cycle, the Software Conventions and Guidelines Document will be developed to identify required, encouraged, discouraged, and forbidden coding techniques. The SST shall have a scope of verification of this document in order to determine the impact of these practices on software safety.

4.3.1.12 Security Analysis

The SST shall have a scope of verification of the design phase documents to confirm that security threats to safety-related software systems have been considered and documented, including;

1. that the architecture correctly handle identified security threats (from the requirements phase), and introduce no new security threats,
2. that the design (in the SDS) ensures unauthorized changes be prevented, detected, or mitigated as appropriate, and
3. that the source code introduce no new security threats into the safety system software.

4.3.2 Input Documents

Inputs to SST design phase activities are defined as follows:

1. Software Design Specification (SDS)
2. Validation Test Procedures and Test Cases
3. Software Conventions and Guidelines Document
4. Internal Data Communication Specifications

4.3.3 Outputs

All SST activities in this phase cumulate with verification activities. Results of all verification activities shall be documented in the verification package.

4.3.4 Methods

Functional Analysis, Logic Analysis, Data Analysis, Constraint Analysis, Timing and Sizing Analysis and Software Element Analysis will be performed by verification of the Software Design Specifications (SDS). The Validation Test Procedures will be verified against the HSS, User's Manual and Software Validation Test Plan as defined in the SMP. Criticality Levels will be established in the SDS analysis. Verification methods are described in the SVVP.

4.3.5 Reporting Requirements

Reporting shall be done in accordance with verification reporting requirements specified in the SVVP and EOP 42-6.0.

4.4 Implementation Phase Analysis

4.4.1 Task Description

In this phase of development, each code module will be reviewed to ensure that the code adequately addresses each of the requirements and hazards. The SST shall have a scope of verification of the output documents in the Implementation phase to confirm the analysis described in this section.

4.4.1.1 Logic Analysis

Logic analysis evaluates the sequence of operations represented by the coded program and detects programming errors that might create hazards. Module testing is generally

used as a tool in the performing logic analysis. Any errors would generally be corrected prior to release of the code.

4.4.1.2 Data Analysis

Data analysis evaluates the data structure and usage in the code to ensure each is defined and used properly by the program. Analysis of data items used by the program is usually performed in conjunction with the logic analysis. The SST shall have a scope of the of code review to verify data structures of safety software and to ensure that the data structures do not introduce new hazards.

4.4.1.3 Interface Analysis

Interface analysis ensures compatibility of the program modules with each other and with external hardware and software. Testing of the interfaces between program modules is conducted in the Integration Testing Phase of the plant life cycle. The SST shall have a scope of verification of the integration testing report to confirm that test cases and boundary conditions have been defined and tested for all interfaces between modules as well any hardware or external software systems in safety system software.

4.4.1.4 Constraint Analysis

Constraint analysis ensures that the program operates with the constraints imposed upon it by the requirements, the design and the target computer. Constraint analysis is designed to identify these limitations, to ensure that the program operates within them, and to ensure that all interfaces have been considered for out-of-sequence and erroneous inputs. The SST shall have a scope of verification of the module testing report to confirm that test cases and boundary conditions have been defined and tested for all constraints in safety system software.

4.4.1.5 Programming Style Analysis

Programming style analysis ensures that all portions of the program follow approved programming guidelines. This analysis is performed by the VVT during the code verification as specified by the SMP, SVVP and SQAP.

4.4.1.6 Non-critical Code Analysis

Non-critical code analysis examines portions of the code that are not considered safety-critical to ensure that they do not cause hazards. The intent of this analysis is to verify that the isolation as documented by the design is properly implemented and that new hazards are not introduced.

4.4.1.7 Timing and Sizing Analysis

The SST shall have a scope of the code review to verify that timing and sizing requirements developed in earlier phases of the software life cycle have been correctly

considered in the implementation and to ensure that no hazards due to timing or sizing factors have been added by the process of writing the code.

4.4.2 Input Documents

Inputs to SST Implementation Phase activities are defined as follows:

1. Source Code
2. Module Test Reports

4.4.3 Outputs

All SST activities in this phase cumulate with verification activities. Results of all verification activities shall be documented in the verification package.

4.4.4 Methods

Verification methods are described in the SVVP.

4.4.5 Reporting Requirements

Reporting shall be done in accordance with verification reporting requirements specified in the SVVP and EOP 42-6.0.

4.5 Integration Test Evaluation

4.5.1 Task Description

The SVVP and SIntP define the software testing activities, including stress testing and regression testing, that will be carried out as part of the Integration Testing Phase of software development. The Integration Testing in combination with the Validation Testing performed in the next phase of the life cycle should provide testing coverage for all software safety requirements. No specific testing will be performed to support the SST. Any deficiencies in the test programs will be addressed by revisions to the appropriate test requirements documents. The SST shall have a scope of verification of integration test reports for safety systems employing software verify the traceability analysis provided covers test of all safety features.

Recommendations for additional integration testing will be provided if necessary.

4.5.2 Input Documents

Inputs to SST Integration Phase activities are defined as follows:

1. Integration Test Reports

4.5.3 Outputs

All SST activities in this phase cumulate with verification activities. Results of all verification activities shall be documented in the verification package.

4.5.4 Methods

Verification methods are described in the SVVP.

4.5.5 Reporting Requirements

Reporting shall be done in accordance with verification reporting requirements specified in the SVVP and EOP 42-6.0.

4.6 Validation Test Evaluation

4.6.1 Task Description

The SMP defines the software testing activities, including stress testing and regression testing that will be carried out as part of the Validation Testing Phase of software development. This section describes how the results of the validation testing will be used to show testing coverage for software safety requirements. The Validation Testing in combination with the Integration Testing, performed in the previous phase of the life cycle, should provide testing coverage for all software safety requirements. No specific testing will be performed to support the verification of SSA activities. Any deficiencies in the test programs will be addressed by revisions to the appropriate test requirements documents.

The SST shall verify the validation test plans and validation test reports for safety systems to confirm that the SSA activities have been performed as defined by the SVVP.

Recommendations for additional testing will be provided if necessary. All software safety requirements must be demonstrated through at least one testing method.

4.6.2 Input Documents

1. Validation Test Report

4.6.3 Outputs

Verification reporting as required by EOP 42-6.0. Including verification of the traceability matrix documenting the relationship between the validation testing and the safety software requirements. The verification should include documentation of any recommendations for additional testing.

4.6.4 Methods

Verification methods are described in the SVVP.

4.6.5 Reporting Requirements

Reporting shall be done in accordance with verification reporting requirements specified in the SVVP and EOP 42-6.0.

4.7 Task Iteration

Changes made to the safety documentation shall follow the task iteration policy of the

SMP, SVVP, and SQAP, and shall include Software Safety Organization tasks.

5 Post Development

The training, deployment, monitoring, maintenance, and retirement of Safety-related Software necessary to ensure the continued safety of the plant during plant operation are beyond the scope of the supplier. Therefore, the planning for these activities shall be developed separately as discussed below.

5.1 Training

The operator is responsible for the training requirements for the safe operation and use of the plant including I&C.

5.2 Deployment

This section of the SSP defines the installation, operations support, and startup requirements of the safety-critical software.

5.2.1 Installation

Safety-related software is installed in the equipment as firmware by the vendor in accordance with established procedures. Therefore, no software installation documentation will be provided for safety-related software. A software safety evaluation shall be performed to ensure that the procedures identify all steps necessary to configure the software for the particular application and that appropriate documentation is completed to provide an accurate record of the installation process.

Installation of nonsafety-related software shall be described in the Operations and Maintenance Manual and is outside the scope of this plan.

A Software Safety Analysis of all factory acceptance test (FAT) plans will be performed to ensure that insights from previous phases of the Safety Analysis are adequately addressed in the FAT program. Any changes required to the software as a result of the FAT shall be evaluated in accordance with the procedures in Section 4.7 of this plan.

5.2.2 Startup and Transition

The startup requirements will be established by the Joint Test Group as described in the DCD. Transition is not applicable to this project since there is no existing system.

5.2.3 Operations Support

Users manuals shall be provided for all systems and instruments in accordance with the SMP.

5.3 Monitoring

The plant operator is responsible for monitoring the operation of the safety-critical software within the system and shall identify procedures for documenting and reporting all safety concerns that are detected during its operation.

If GE identifies a condition, which could have implications for safety-critical software, the operator will be notified in accordance with GE Policy and Procedure 70-42.

5.4 Maintenance

Any maintenance of the software during plant start up shall be performed in accordance with the SMP. Maintenance of the software is after plant startup is beyond the scope of supply of the vendor.

5.5 Retirement and Notification

Retirement and notification are beyond the scope of this plan.

6 Plan Approval

Plan approval is performed in accordance with EOPs.

Table 1 Software Life Cycle

Life Cycle Phase	Software Safety Analysis Activities
Planning	SSP Development Analysis Preparation
Requirements	Requirements Analysis
Design	Design Analysis
Implementation	Code Evaluation Test Analysis
Integration	Test Analysis
Validation	Test Analysis
Installation	N/A
Operations and Maintenance	N/A