ANP-10272
Revision 1

Software Program Manual for TELEPERM XS™ Safety Systems

Topical Report

XXXXXX 2009

AREVA NP Inc.

Non-Proprietary

(c) 2009 AREVA NP Inc.

Draft

Copyright © 2009

AREVA NP Inc.
All Rights Reserved

The design, engineering, and other information contained in this document have been prepared by or on behalf of AREVA NP Inc., a jointly owned subsidiary of AREVA and Siemens, in connection with customer requests to use the TELEPERM XS system in U.S. power plants. No use of or right to copy any of this information, other than by the NRC and its contractors in support of AREVA NP's pre-application review, is authorized.

The information provided in this document is a subset of a much larger set of know-how, technology, and intellectual property pertaining to the TELEPERM XS system. Without access and a grant of rights to that larger set of know-how, technology and intellectual property rights, this document is not practically or rightfully usable by others, except by the NRC as set forth in the previous paragraph.

For information address: AREVA NP Inc.
An AREVA and Siemens Company
3315 Old Forest Road
Lynchburg, VA 24506

Draft

U.S. Nuclear Regulatory Commission

Disclaimer

Important Notice Concerning the Contents and Application of This Report

Please Read Carefully

This report was developed based on research and development funded and conducted by AREVA NP Inc., and is being submitted by AREVA NP to the U.S. Nuclear Regulatory Commission (NRC) to facilitate future licensing processes that may be pursued by licensees or applicants that are customers of AREVA NP. The information contained in this report may be used by the NRC and, under the terms of applicable agreements with AREVA NP, those customers seeking licenses or license amendments to assist in demonstrating compliance with NRC regulations. The information provided in this report is true and correct to the best of AREVA NP's knowledge, information, and belief.

AREVA NP's warranties and representations concerning the content of this report are set forth in agreements between AREVA NP and individual customers. Except as otherwise expressly provided in such agreements with its customers, neither AREVA NP nor any person acting on behalf of AREVA NP:

- Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, nor the use of any information, apparatus, method, or process disclosed in this report.
- Assumes any liability with respect to the use of or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

Draft

ABSTRACT

This Software Program Manual describes the program measures incorporated at AREVA NP to:

- Ensure that the TELEPERM XS Application Software attains a level of quality commensurate with its importance to safety functions
- Ensure that the Application Software performs the required safety functions correctly
- Conform to established technical and documentation requirements, conventions, rules, and industry standards

This Software Program Manual describes the requirements and objectives for the following plans:

1. Software Management Plan, which describes the overall management process used for the development of project-specific TELEPERM XS Application Software.
2. Software Development Plan, which describes the life cycle activities for TELEPERM XS Application Software development.
3. Software Quality Assurance Plan, which describes the necessary processes that ensure that the software attains a level of quality commensurate with its importance to safety function.
4. Software Integration Plan, which describes the software integration process and the hardware/software integration process for TELEPERM XS projects.
5. Software Installation Plan, which describes the installation process for TELEPERM XS projects.
6. Software Operations and Maintenance Plan, which describes post-customer delivery TELEPERM XS Ssoftware practices.
7. Software Training Plan, which describes a process that can be used to ensure that training needs of appropriate plant staff, including operators and I&C engineers and technicians, are met.

Draft

8. Software Safety Plan, which identifies the process to reasonably eliminate hazards that could jeopardize the health and safety of the public from safety-critical software.
9. Software Verification and Validation Plan, which describes the method that ensures correctness of the TELEPERM XS Application Software.
10. Software Configuration Management Plan, which describes the method that maintains the project-specific TELEPERM XS Software in a controlled configuration.
11. Software Test Plan, which describes the purpose and scope of the TELEPERM XS Application Software testing activities.

The Software Program Manual, which defines the plans listed above, is implemented through AREVA NP Operating Instructions, and constitutes a program that conforms to applicable Nuclear Regulatory Commission guidance.

Draft

Nature of Changes

Revision	Section(s) or Page(s	Description and Justification
0	All	Initial issue.
1	All	Incorporated information from NRC requests for additional information.

Draft

Contents

Page

1.0 INTRODUCTION 1-1
2.0 DEFINITIONS 2-1
3.0 SOFTWARE MANAGEMENT PLAN 3-1
3.1 Key Interfaces 3-2
3.1.1 Interface with AREVA NP Quality Assurance Programs 3-2
3.1.2 Interface with the TELEPERM XS Topical Report 3-4
3.1.3 Use of TELEPERM XS Technology 3-5
3.1.4 Relationship to BTP 7-14 Guidance 3-6
3.1.5 Relationship to Cyber Security Guidance 3-9
3.1.6 Project-Specific Actions Items 3-9
3.2 Organization 3-10
3.2.1 Roles and Responsibilities 3-11
3.3 TELEPERM XS Application Software Development Controls 3-16
3.4 Problem Reporting 3-17
3.4.1 Corrective Action Program 3-17
3.4.2 Open Items 3-18
3.4.3 Discrepancies Identified by Design Reviews and Audits 3-19
3.4.4 Discrepancies Identified by Testing 3-19
3.4.5 Discrepancies Identified by Verification and Validation 3-20
3.4.6 Discrepancies Identified after Release to the Customer 3-20
4.0 SOFTWARE DEVELOPMENT PLAN 4-1
4.1 Use of TELEPERM XS Technology 4-1
4.2 TELEPERM XS Project-Specific Development Process Overview 4-3
4.3 TELEPERM XS Application Software Life Cycle 4-6
4.4 Software Classification 4-11
4.5 Software Development Documentation 4-11
4.5.1 System Design Requirements Document 4-12
4.5.2 System Design Description 4-12
4.5.3 Software Requirements Specification 4-12
4.5.4 Software Design Description 4-14
4.5.5 Application Software Requirements Traceability Matrix 4-14
4.5.6 Test Plan 4-15
4.5.7 Test Documentation 4-15
4.6 Security and Disaster Recovery 4-16
4.6.1 Cyber Security Design Features 4-16
4.6.2 Cyber Security Administrative Controls 4-18
4.7 Documentation Standards 4-18
4.7.1 Naming Conventions 4-18
4.7.2 Coding Standards 4-18
4.7.3 Logic Structure Standards 4-19

Draft

4.7.4 Function Diagram Standards 4-19
4.7.5 Code Configuration 4-19
4.8 User Manuals 4-20
5.0 SOFTWARE QUALITY ASSURANCE PLAN 5-1
5.1 Purpose 5-1
5.2 Management 5-1
5.3 Documentation 5-2
5.4 Software Reviews 5-2
5.4.1 Software Requirements Review 5-3
5.4.2 Preliminary Design Review 5-3
5.4.3 Detailed Design Review 5-3
5.4.4 Software Verification and Validation Plan Review 5-4
5.4.5 Managerial Reviews 5-4
5.4.6 Software Configuration Management Plan Review 5-4
5.4.7 Post-implementation Review 5-4
5.5 Software Audits 5-4
5.5.1 In-Process Audits 5-5
5.5.2 Physical Audits 5-5
5.5.3 Functional Audits 5-5
5.5.4 Software Process Audits 5-5
5.6 Testing 5-6
5.6.1 Test Planning 5-6
5.6.2 Test Specifications 5-6
5.6.3 Test Reporting 5-7
5.7 Standards 5-7
5.8 Problem Reporting and Corrective Action 5-7
5.9 Tools, Methodologies, and Metrics 5-7
5.9.1 Tools and Methodologies 5-7
5.9.2 Metrics 5-9
5.10 Media Control 5-9
5.10.1 Media Control for Application Software 5-10
5.10.2 Code Control of TELEPERM XS System Software 5-10
5.11 Supplier Control 5-10
5.11.1 Software Development by Third Parties 5-10
5.11.2 Existing Software Developed by Third Parties 5-10
5.12 Records Collection, Maintenance, and Retention 5-11
5.13 Training 5-11
5.14 Risk Management 5-11
5.14.1 Risk Management Process 5-12
5.14.2 Independent Risk Analysis 5-12
5.14.3 Standard TELEPERM XS Risk Mitigation Measures 5-13
6.0 SOFTWARE INTEGRATION PLAN 6-1
7.0 SOFTWARE INSTALLATION PLAN 7-1
8.0 SOFTWARE MAINTENANCE AND OPERATIONS PLAN 8-1

Draft

8.1 Purpose 8-1
8.2 Problem Identification 8-2
8.3 Analysis 8-4
8.4 Processing Simple Changes without Nonconformances 8-4
9.0 SOFTWARE TRAINING PLAN 9-1
9.1 Purpose 9-1
9.2 Organization 9-3
9.3 Responsibilities 9-4
9.4 Measurements 9-4
9.5 Procedures 9-5
9.6 Methods 9-5
9.7 Training Manuals and Materials 9-6
10.0 SOFTWARE SAFETY PLAN 10-1
10.1 Purpose 10-1
10.2 Management 10-1
10.3 Software Safety Analyses 10-3
10.3.1 Preliminary Hazard Analysis 10-4
10.3.2 Diversity and Defense-in-Depth Analysis 10-5
10.3.3 Application Software Requirements Traceability Analysis 10-5
10.3.4 Failure Modes and Effects Analysis 10-6
10.3.5 Response Time Analysis 10-7
10.3.6 Verification and Validation Activities 10-8
10.3.7 Application Software Validation Testing 10-8
10.3.8 Criticality Analysis 10-9
10.3.9 System Testing 10-9
10.4 Documenting and Correcting Safety Hazards 10-10
11.0 SOFTWARE VERIFICATION AND VALIDATION PLAN 11-1
11.1 Purpose 11-3
11.2 Overview 11-3
11.2.1 Organization 11-3
11.2.2 Procedures 11-5
11.2.3 Schedule 11-5
11.2.4 Software Integrity Levels 11-6
11.2.5 Resources 11-6
11.2.6 Responsibilities 11-6
11.2.7 Verification and Validation Methods 11-8
11.2.8 Independent Testing and Validation 11-17
11.2.9 Regression Testing 11-17
11.3 Metrics 11-17
11.3.1 Metrics for Software Development Effectiveness 11-18
11.3.2 Metrics for Verification and Validation Effectiveness 11-18
11.4 Verification and Validation Reports 11-19
12.0 SOFTWARE CONFIGURATION MANAGEMENT PLAN 12-1

Draft

12.1 Introduction 12-1
12.1.1 Purpose 12-1
12.1.2 Scope 12-1
12.2 Management 12-2
12.2.1 Organization 12-2
12.2.2 Responsibilities 12-2
12.2.3 Configuration Control Boards 12-4
12.3 Software Configuration Management Implementation Activities 12-5
12.3.1 Configuration Item Naming and Labeling 12-5
12.3.2 Protection of Configuration Items 12-6
12.3.3 Purchased Software 12-7
12.3.4 Software Library 12-8
12.3.5 Configuration Baseline Management 12-8
12.3.6 Change Management and Configuration Control 12-9
12.3.7 Modification Procedures 12-10
12.3.8 Software Download Control 12-10
12.3.9 Modifying Changeable Parameters 12-11
12.3.10 Problem Reporting 12-12
12.3.11 Status Accounting, Reviews and Audits 12-12
13.0 SOFTWARE TEST PLAN 13-1
13.1 Alignment with IEEE Std 1012-1998 Testing Activities 13-2
13.2 Application Software Validation Testing with Simulation Test Tool 13-3
13.3 System Validation Testing in the Test Field 13-5
13.3.1 System Validation Test Scope 13-7
13.3.2 Test Field Validation Tests (System and Acceptance Tests) 13-9
13.3.3 System Validation Test Documents 13-14
13.3.4 Test Reporting 13-15
13.3.5 Summary of Test Field Validation Testing 13-17
14.0 CONCLUSIONS 14-1
15.0 REFERENCES 15-1
15.1 U.S. Regulations 15-1
15.2 U.S. Regulatory Guidance 15-1
15.3 U.S. Industry Standards 15-2
15.4 Regulatory Review Precedent 15-3
15.5 AREVA NP Documents 15-3

Draft

List of Tables

Page

Table 1-1 AREVA NP Alignment with BTP 7-14 1-2

Table 13-1 Alignment with IEEE Std 1012-1998 Test Activities..... 13-4

List of Figures

Page

Figure 3-1 - Annotated Version of Figure 2.2 from the TELEPERM XS Topical Report 3-5

Figure 3-2 - TELEPERM XS Software Program Manual Scope 3-7

Figure 3-3 - Annotated Version of BTP 7-14 Figure 7-A-1 3-8

Figure 11-1 - Test Document Work Flow and Control Points..... 11-2

Appendices

Page

Appendix A - Map of AREVA NP Software Life Cycle Activities to IEEE Std 1074 SoftwareLife Cycle A-1

Appendix B - Historical Information on SPACE Tool Qualification..... B-1

Draft

Nomenclature

Acronym	Definition
ASL	Approved Supplier List
BTP	Branch Technical Position
CFR	Code of Federal Regulations
CPU	Central Processing Unit
CRC	Cyclic Redundancy Checksum
ERBUS	TELEPERM XS computer-assisted test system for TELEPERM XS test field application (test field simulator)
FAT	Factory Acceptance Test
FB	Function Block
FD	Function Diagram
FDG	Function Diagram Group
FDGM	Function Diagram Group Module
FMEA	Failure Modes and Effects Analysis
GSM	Graphic Service Monitor
HICB	Human, Instrumentation and Controls Branch
I&C	Instrumentation and Control
I/O	Input/Output
ID	Identification
IEEE	Institute of Electrical and Electronic Engineers
NRC	Nuclear Regulatory Commission
NUPIC	Nuclear Procurement Issues Committee
QA	Quality Assurance
RTE	Runtime Environment
SDD	Software Design Description
SDRD	System Design Requirements Document
SIL	Software Integrity Level
SME	Subject Matter Expert
SPACE	Specification and Coding Environment
SRS	Software Requirements Specification
SysDD	System Description Document

Draft

1.0 INTRODUCTION

This Software Program Manual for TELEPERM XS Safety Systems (hereafter called the Software Program Manual) describes the program measures incorporated by AREVA NP Inc. (AREVA NP) to:

- Ensure that the TELEPERM XS Application Software attains a level of quality commensurate with its importance to safety functions
- Ensure that the Application Software performs the required safety functions correctly
- Conform to established technical and documentation requirements, conventions, rules, and industry standards

The Software Program Manual establishes the requirements and objectives for the various software plans. These plans are implemented as AREVA NP Operating Instructions and conform to the requirements established in the Software Program Manual. Operating Instructions are used to define specific implementation details. For example, the Software Configuration Management Plan is defined in an Operating Instruction and additional administrative controls for the software library are specified in a separate Operating Instruction. Operating instructions established for these plans are available onsite at AREVA NP facilities to support NRC review of this topical report. Table 1-1 shows how the suggested plans from BTP 7-14 are addressed.

AREVA NP requests that the NRC issue a Safety Evaluation Report that approves the use of this Software Program Manual. AREVA NP intends to use the Software Program Manual to support digital safety instrumentation and control (I&C) system upgrades at operating nuclear plants and digital safety systems for new nuclear plants. For instance, the approved version of this topical report would be referenced in the Design Control Document for the U.S. EPR.

Draft

Table 1-1 AREVA NP Alignment with BTP 7-14

Suggested Plan from BTP 7-14	Section of this Report in which Discussed
Software Management Plan	3.0
Software Development Plan	4.0
Software Quality Assurance Plan	350
Software Integration Plan	6.0
Software Installation Plan	7.0
Software Maintenance Plan	8.0
Software Training Plan	9.0
Software Operations Plan	8.0
Software Safety Plan	10.0
Software Verification and Validation Plan	11.0
Software Configuration Management Plan	12.0
Software Test Plan	13.0

Draft

2.0 DEFINITIONS

Anomaly [IEEE Std 1012]

Any condition that deviates from the expected condition based on requirements, specification, design, documents, user documents standards, or from someone's perceptions or experiences. Anomalies may be found during but are not limited to, the review, test, analysis, compilation, or use of software products or applicable documentation.

Application Software

The Application Software reflects the plant specific functionality of the TELEPERM XS I&C system. It is documented and generated by the TELEPERM XS SPACE tool. The platform system software uses this configuration data to carry out the application specific functionality of the I&C system.

Baseline [IEEE Std 610.12 (Reference 19)]

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. Formal review and agreement means that responsible management has reviewed and approved a baseline. Baselines are subject to change control.

Baseline Management [IEEE Std 610.12]

In configuration management, the application of technical and administrative direction to designate the documents and changes to those documents that formally identify and establish baselines at specific times during the life cycle of a configuration item.

Draft

Changeable Parameters

Parameters used to operate the unit from external devices in order to perform functions. Examples include inserting revised setpoints and bypasses or performing periodic testing.

Code [IEEE Std 610.12]

Computer instructions and data definitions expressed in a programming language or in a form output by an assembler, compiler, or another translator.

Component [IEEE Std 610.12]

One of the parts that make up a system. A component may be hardware or software and may be subdivided into other components.

Configuration [IEEE Std 828 (Reference 21)]

The arrangement of a computer system or component as defined by the number, nature, and interconnections of its constituent parts. In configuration management, the functional and physical characteristics of hardware or software as set forth in technical documentation or achieved in a product.

Configuration Control [IEEE Std 610.12]

An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification.

Configuration Control Board [IEEE Std 610.12]

A group of people responsible for evaluating and approving or disapproving proposed changes to configuration items, and for ensuring implementation of approved changes.

Draft

Configuration Identification [IEEE Std 610.12]

An element of configuration management, consisting of selecting the configuration items for a system and recording their functional and physical characteristics in technical documentation.

The current approved technical documentation for a configuration item as set forth in specifications, drawings, associated lists, and documents referenced therein.

Configuration Item [IEEE Std 610.12]

An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process.

Configuration Management [IEEE Std 610.12]

A discipline applying technical and administrative direction and surveillance to:

- Identify and document the functional and physical characteristics of a configuration item
- Control changes to those characteristics
- Record and report change processing and implementation status
- Verify compliance with specified requirements

Configuration Status Accounting [IEEE Std 610.12]

An element of configuration management, consisting of the recording and reporting of information needed to manage a configuration effectively. This information includes a listing of the approved configuration identification, the status of proposed changes to the configuration, and the implementation status of approved changes.

Draft

Control Point [IEEE Std 828]

A project agreed upon point-in-time when specified agreements or controls are applied to the software configuration items being developed, such as an approved baseline or release of a specified document/code.

Coverage

Method and indicators to assess that the functional features of the software have been comprehensively validated.

cpuload

TELEPERM XS load analysis tool used to analyze the loading on the central processor units.

Cyclic Redundancy Checksum (CRC)

Method applied for identification of data files using industry standard functions to produce a unique checksum. This checksum is used to identify and detect alteration of data during usage, transmission, or storage.

Design Review Board

A board of knowledgeable personnel that review the design for a product to assure that the introduction of new products, new processes, significant changes to existing products or processes, corrective actions for failed products or processes, or any other projects judged to warrant a design review, result in the delivery of high quality and reliable products to the customer.

Discrepancies

During the software development life cycle, any difference or perceived difference discovered by various organizations in the later documents or code with the earlier requirements found in the customer's specification, the SDRD or the SRS. These

Draft

discrepancies are initially documented on the Open Item list and are evaluated for further action.

Electrically Erasable Programmable Read-Only Memory (EEPROM)

Type of non-volatile memory used in computers and other electronic devices to store small amounts of data that must be saved when power is removed.

Flash Erasable Programmable Read-Only Memory (FEPROM)

Type of non-volatile computer memory that can be electrically erased and reprogrammed in large blocks.

FunBase

A tool that administrates the naming of software modules, parameters, signals, data tables and other entities in the specification or design so that each entity is uniquely and consistently named and properly connected in the Application Software.

Functional Configuration Audit [IEEE Std 610.12]

An audit conducted to verify that the development of a configuration item has been completed satisfactorily, that the item has achieved the performance and functional characteristics specified in the functional or allocated configuration identification, and that its operational and support documents are complete and satisfactory.

Interface [IEEE Std 610.12]

1. A shared boundary across which information is passed. This boundary includes design interfaces between design organizations (as interpreted by Regulatory Guide 1.169).
2. A hardware or software component that connects two or more other components for the purpose of passing information from one to the other.

Draft

3. To connect two or more components for the purpose of passing information from one to the other.
4. To serve as a connecting or connected component as in 2 above.

Interface Control [IEEE Std 610.12]

In configuration management, the process of:

- Identifying functional and physical characteristics relevant to the interfacing of two or more configuration items provided by one or more organizations
- Ensuring that proposed changes to these characteristics are evaluated and approved prior to implementation

MIC File

Machine language loadable code file.

netload

TELEPERM XS load analysis tool used to analyze the loading on the network connections.

Open Item

Any item which constitutes an error or anomaly from the required status or condition of a properly completed project. Each Open Item is given an identifier that is unique to the project and unit as well as a record in a database. The entry contains information to track the life cycle of the item from initiation to final resolution.

Physical Configuration Audit [IEEE Std 610.12]

An audit conducted to verify that a configuration item, as built, conforms to the technical documentation that defines it.

Draft

rediff

TELEPERM XS tool used to detect differences in the functionality of Application Software in the redundant divisions of an I&C system. The tool performs an analysis of logics and parameter data specified for redundant system trains and identifies differences in functionality. The differences must be evaluated by an engineer to determine whether the differences are planned (engineered differences) or unplanned (errors).

reflist

A software program that creates cyclic redundancy check (CRC) sums recursively for the subdirectories and files within a directory and outputs them in a list, including the date of the last change for the file. This method is used for identification of the TELEPERM XS system software, for project specific additions, for the Application Software implemented on an engineering platform (engineering workstation), and for software downloaded into the I&C system.

Regression Testing [IEEE Std 610.12]

Selective retesting of a system or component to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements.

scanmic

‘scanmic’ is a TELEPERM XS software authentication tool. It analyzes the software configuration of loadable code (called MIC files). ‘scanmic’ is used to read the version strings of the Application Software components contained in a loadable MIC file from the MIC file itself, and calculate the CRC checksum for each software segment in the MIC file as well as the CRC checksum for the entire MIC file.

Draft

This information can be output to a list which serves to document the generated software version. Differences in the software configuration between the old version and the new version can be determined from these lists and then verified.

Software [IEEE Std 610.12]

Computer programs, procedures, and in some cases, associated documentation and data pertaining to the operation of a computer system.

Software Design Description [IEEE Std 610.12]

A representation of software created to facilitate analysis, planning, implementation, and decision making. The software design description is used as a medium for communicating software design information, and may be thought of as a blueprint model of the system.

Safety Goal

Provide reasonable assurance that the TELEPERM XS Application Software performs its design basis safety function. These design basis functions are described in the SDRD for a project.

Software Risk [Based on IEEE Std 1228 (Reference 31)]

A measure that combines both the likelihood that a software hazard will cause a problem and the severity of that problem. A software risk is minimized in the design process by using the TELEPERM XS SPACE tool, software and system testing, and FAT. The risk is further minimized through the verification and validation process.

Software Hazard [Based on IEEE Std 1228]

A software design error that could lead to an unintended operation or failure to operate when required. The analyses and tests defined in Section 10.3 are specified to provide reasonable assurance that software hazards are eliminated.

Draft

Software Error

An actual mistake in the software. The use of the TELEPERM XS object–oriented automated code generation tools minimizes the inherent risk in the development of the Application Software as well as minimizes the potential for human error. These tools support the development of high quality software.

Software Library [IEEE Std 610.12]

A controlled collection of software and related documentation designed to aid in software development, use, or maintenance. Types include master library, production library, software development library, software repository, and system library.

Software Life Cycle [IEEE Std 610.12]

The period of time that begins when a software product is conceived and ends when the software is no longer available for use.

Software Life Cycle Phases

The following phases make up the safety system software life cycle:

1. Basic Design
2. Detailed Design
3. System Integration and Testing
4. Installation and Commissioning
5. Final Documentation

SPACE

The SPACE engineering system comprises the tools used for the engineering and maintenance of the TELEPERM XS I&C software. In this context, engineering refers to the overall process of creating and testing the Application Software:

- Specification of the I&C functions and hardware topology
- Automatic code generation

Draft

- Software authentication, using reflist and scanmic
- Software loading
- Load analysis tool
- Database administration

System Description Document (SysDD)

A document that describes the design of the system. The SysDD describes the overall system architecture and allocation of the functional requirements within the architecture. This document serves as the basis for the SRS.

System Design Requirements Document (SDRD)

A document that specifies, in detail, the requirements of the system to be installed (new or replaced). The SDRD includes both functional and non-functional requirements. This document may be called by a different name. The information contained within this document may be derived from a separate specification produced by the customer, or this document may be jointly developed by the customer and AREVA NP. This document can be called by a different name, as long as it contains all applicable requirements for the system.

System Software [IEEE Std 610.12]

Software designed for a specific computer system or family of computer systems to facilitate the operation and maintenance of the computer system and associated programs such as operating systems, compilers, and utilities.

Test Plan [IEEE Std 610.12]

A document describing the scope, approach, resources, and schedule of intended test activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning.

Draft

TELEPERM XS Project Phases

The following phases make up the TELEPERM XS project:

1. Elaboration of the quotation
2. Project Start-Up
3. Basic Design
4. Detailed Design
5. Manufacturing
6. System Integration and Testing
7. Installation and Commissioning
8. Final Documentation

TELEPERM XS Project Basic Design Phase

The activities that produce the basic design and functional requirements for the project.

TELEPERM XS Project Detailed Design Phase

The activities that produce a completely specified I&C system that has been validated with Application Software integration and functional tests.

TELEPERM XS System Integration and Testing Phase

Activities during the Application Software production process necessary to assemble and integrate the complete system and perform required testing. These are the primary software design activities wherein system performance is checked and documented to ensure that the required functions are implemented correctly and completely.

Unit [IEEE Std 610.12]

1. A separately testable element specified in the design of a computer software component.

Draft

2. A logically separable part of a computer program.

3. A software component that is not subdivided into other components.

Validation [IEEE Std 610.12]

The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. *Contrast with:* verification.

Verification [IEEE Std 610.12]

The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. *Contrast with:* validation.

Version [IEEE Std 610.12]

An initial release or re-release of a computer software configuration item that is associated with a complete compilation or recompilation of the computer software configuration item.

Verification and Validation [IEEE Std 610.12]

The process of determining whether the requirements for a system or component are complete and correct, the products of each development phase fulfill the requirements or conditions imposed by the previous phase, and the final system or component complies with specified requirements.

Draft

3.0 SOFTWARE MANAGEMENT PLAN

The Software Management Plan describes the overall management process used for the development of project-specific TELEPERM XS Application Software.

AREVA NP projects to implement TELEPERM XS systems are performed in accordance with procedural requirements that cover the system, hardware, and software activities associated with delivering TELEPERM XS systems. This program manual is a subset of those procedural requirements and covers the requirements definition, detailed design, and integration and test phases for the TELEPERM XS Application Software and supporting software created by AREVA NP. It outlines the process to be followed for the TELEPERM XS Application Software. The customer may perform some of the identified activities. Project-specific exceptions to the provisions of this plan are documented and justified in the specific project plan.

This Software Program Manual applies to safety-related TELEPERM XS system applications. In addition, it includes certain non-safety application software, such as Graphic System Monitor software and any TELEPERM XS Gateway and interface software.

AREVA NP purchases certain safety-related products that are developed by AREVA NP GmbH under its quality assurance (QA) program. AREVA NP GmbH is an approved supplier per AREVA NP's approved supplier list (ASL). This Software Program Manual describes the duties and responsibilities of AREVA NP and delegates specific duties to the AREVA NP GmbH quality program where applicable.

Following factory acceptance testing (FAT), the customer's QA program normally controls further work on the system; however, during the operating and maintenance phases of the product life cycle, the customer may request support that includes the development of new versions of the Application Software under the governance of this Software Program Manual.

Draft

3.1 Key Interfaces

The project-specific process for implementing a TELEPERM XS system has several key interfaces. Section 3.1.1 describes the interface with the AREVA NP Quality Assurance Programs. Section 3.1.2 describes the interface with the TELEPERM XS Topical Report. Section 3.1.3 describes the use of TELEPERM XS technology. Section 3.1.4 describes the relationship to BTP 7-14 Guidance. Section 3.1.5 describes the relationship to cyber security guidance. Section 3.1.6 describes the project-specific actions items that must be implemented.

3.1.1 Interface with AREVA NP Quality Assurance Programs

All design work, products, and services provided for a TELEPERM XS project are performed to the requirements of the AREVA NP Quality Management Manual (Reference 37). TELEPERM XS Application Software is also produced in accordance with the requirements defined in this Software Program Manual. The Application Software documents conform to the additional QA requirements defined in the Software Quality Assurance Plan.

AREVA NP's implementation of the Quality Management Manual is periodically audited by the Nuclear Procurement Issues Committee (NUPIC). The NUPIC program evaluates suppliers furnishing safety-related components and services and commercial grade items to nuclear utilities.

The Quality Management Manual also allows for the issuance of QA plans to augment the quality requirements for a specific customer or project. AREVA NP Topical Report ANP-10266A, (Reference 35) was issued to describe the Quality Assurance Plan applicable to the Design Certification of the U.S. EPR. The plan is based on the eighteen point criteria of 10 CFR 50, Appendix B (Reference 2), and ANSI/ASME NQA-1-1994 (Reference 13) and is referred to as the U.S. EPR Quality Assurance Plan. Work performed for the U.S. EPR also complies with the requirements defined in the U.S. EPR Quality Assurance Plan.

Draft

The Software Program Manual describes the program measures incorporated by AREVA NP to ensure that the Application Software attains a level of quality commensurate with its importance to safety functions, performs the required safety functions correctly, and conforms to established technical and documentation requirements, conventions, rules, and industry standards. The Software Program Manual applies to Application Software developed for all TELEPERM XS projects in the U.S., including U.S. EPR projects.

The Software Program Manual requires that a Software Quality Assurance Plan be developed. The Software Quality Assurance Plan describes the necessary measures to make sure that the developed Application Software conforms to established technical requirements, rules, and standards. It also describes the tools to be used and methodology to be followed in developing and maintaining software to be used for the design of Application Software.

TELEPERM XS Application Software elements produced in this process include:

- Test plans, cases, procedures, and reports
- Review and audit results
- Problem reports and corrective action documentation
- Software Configuration Management Plans
- Software Verification and Validation Plans
- Software Safety Plans
- Design Documents
- Application Code

Compliance with this Software Program Manual is ensured through Operating Instructions.

Project documentation used as design input or delivered to the customer as design output is stored in the AREVA NP Records Management System. Similarly, project records arising from QA inspections and audits are stored in the AREVA NP Records

Draft

Management System. The record storage requirements are described in the AREVA NP Records Management Program Manual.

3.1.2 Interface with the TELEPERM XS Topical Report

The TELEPERM XS Topical Report (Reference 34) describes the generic qualification process for the safety-related TELEPERM XS digital I&C system. The TELEPERM XS Topical Report also describes a general framework for the implementation of individual projects using the TELEPERM XS technology described in Section 5.1.3 of the TELEPERM XS Topical Report. As noted in the NRC Safety Evaluation Report for the TELEPERM XS Topical Report (Reference 32), NRC did not address plant-specific Application Software development activities. Instead, each applicant using the TELEPERM XS Topical Report is required to address plant-specific Application Software development activities, as noted in plant-specific action item 2.

Based on the initial experience with implementation of plant-specific TELEPERM XS projects, AREVA NP decided it would be prudent to address many of the Application Software development process issues in a standard way. As such, the TELEPERM XS Software Program Manual augments the generic system qualification process described in the TELEPERM XS Topical Report with a standard engineering process used to develop TELEPERM XS Application Software for U.S. projects. The relationship of the two topical reports is shown in Figure 3-1, Annotated Version of Figure 2.2 from the TELEPERM XS Topical Report.

The Software Program Manual describes the standard engineering process used to develop TELEPERM XS Application Software for U.S. projects. The Software Program Manual, along with the five project-specific action items listed in Section 3.1.6, address plant-specific action items 2 and 17 from the NRC Safety Evaluation Report for the TELEPERM XS Topical Report. Applicants using the TELEPERM XS technology for safety-related projects can satisfy plant-specific action item 2 and 17 by referencing this Software Program Manual and addressing the five project-specific actions items.

Draft

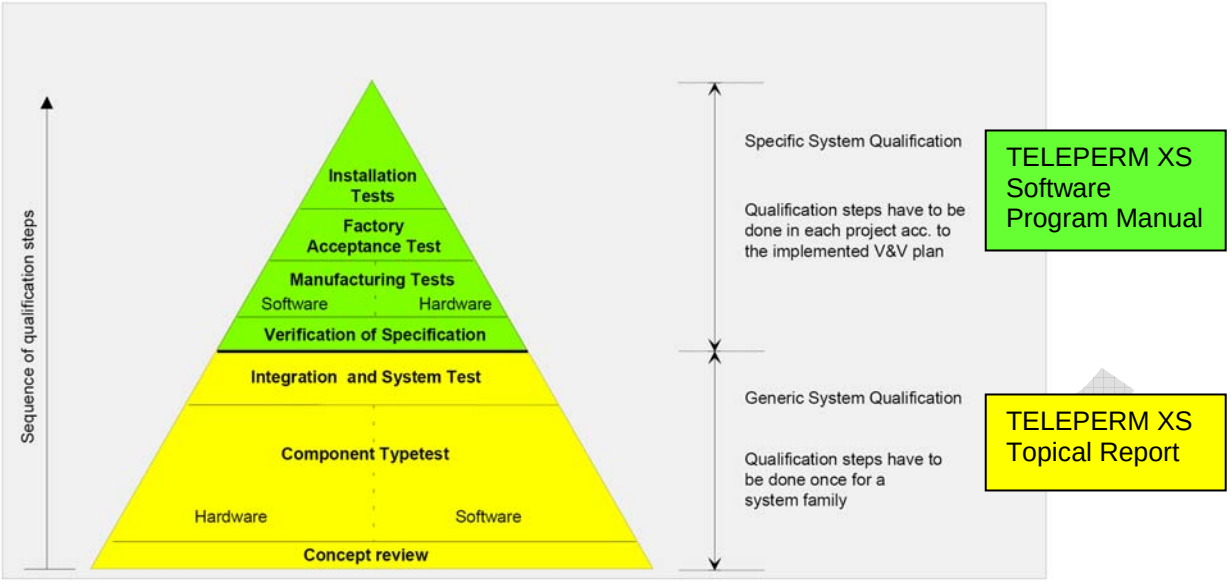


Figure 3-1 - Annotated Version of Figure 2.2 from the TELEPERM XS Topical Report

3.1.3 Use of TELEPERM XS Technology

As noted above, TELEPERM XS Topical Report Section 5.1.3 describes a general framework for the implementation of individual projects using the TELEPERM XS technology. An integral part of that framework is the use of the TELEPERM XS engineering tools set for the development of Application Software. A key feature of the TELEPERM XS tool set is the qualified automatic code generator in the Specification and Coding Environment (SPACE) tool, which was approved as part of the TELEPERM XS Topical Report. The Software Program Manual is predicated on the use of the SPACE tool for code generation. The relationship of the two topical reports with regard to the various TELEPERM XS technology components is shown in Figure 3-2, TELEPERM XS Software Program Manual Scope.

3.1.4 Relationship to BTP 7-14 Guidance

The Software Program Manual describes the engineering processes used to develop TELEPERM XS Application Software. Specifically, it fully addresses the planning and development activities for all Application Software lifecycle group activities and partiallyaddresses those associated with Operations, Maintenance, and Training, from the requirements phase through the validation phase. The scope of this Software Program Manual is shown in Figure 3-3, Annotated Version of BTP 7-14 Figure 7-A-1. Applicants using the TELEPERM XS technology for safety-related projects will need to address the software planning and development activities for the Operations, Maintenance, and Training topics for their scope or responsibility. AREVA NP support for these activities is described in Sections 8.0 and 9.0 of this Software Program Manual.

AREVA NP addresses configuration management during Application Software development. Applicants will need to address software configuration management after system delivery and during operational phases.

AREVA NP validation activities end with the system validation test (including FAT). Subsequent installation activities, starting with the site acceptance test, are the primary responsibility of the customer. AREVA NP support for installation activities are defined by the customer for each project. Applicants using the TELEPERM XS technology for safety-related projects will need to address the software planning and development activities for the Installation phase.

Draft

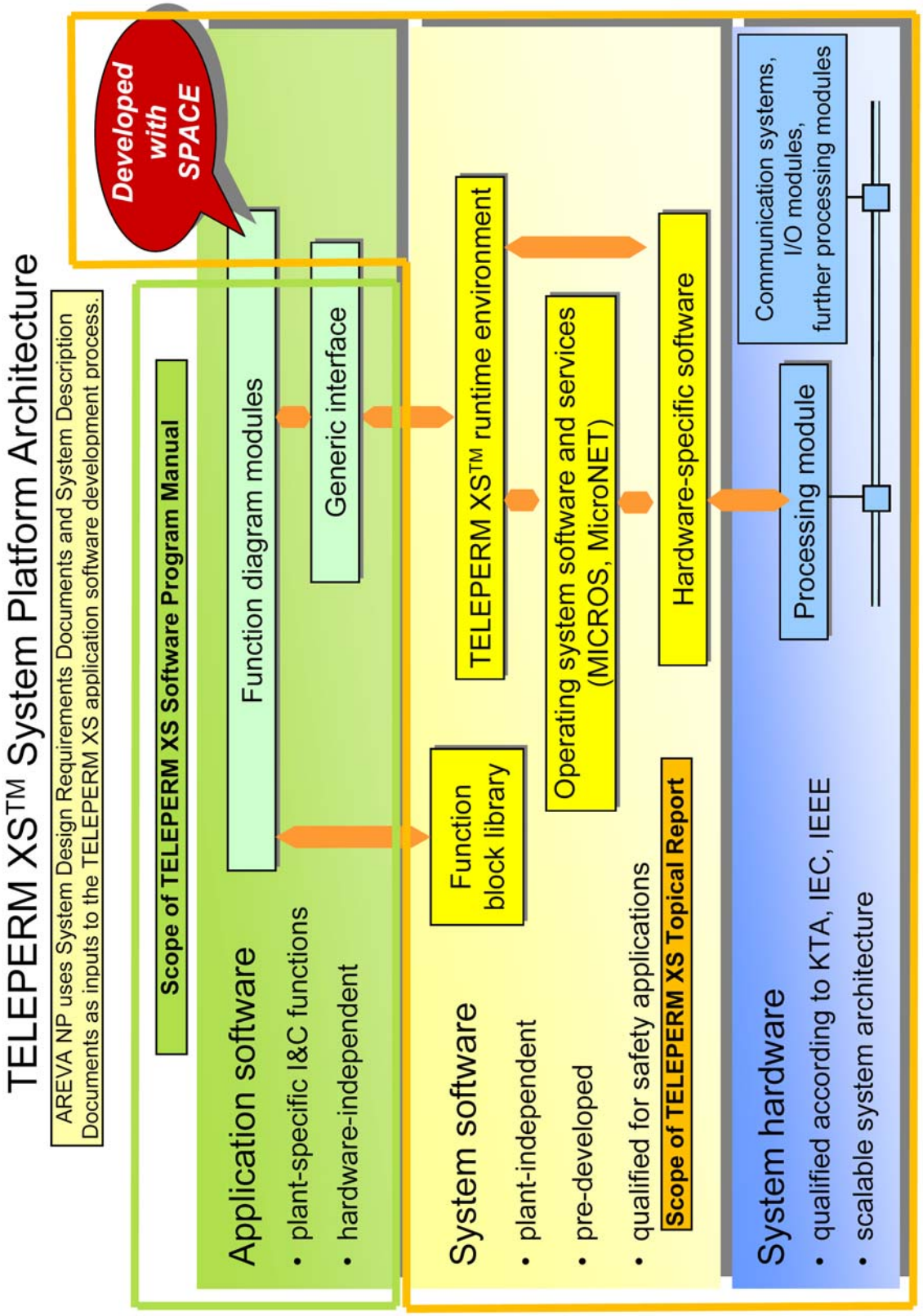


Figure 3-2 - TELEPERM XS Software Program Manual Scope

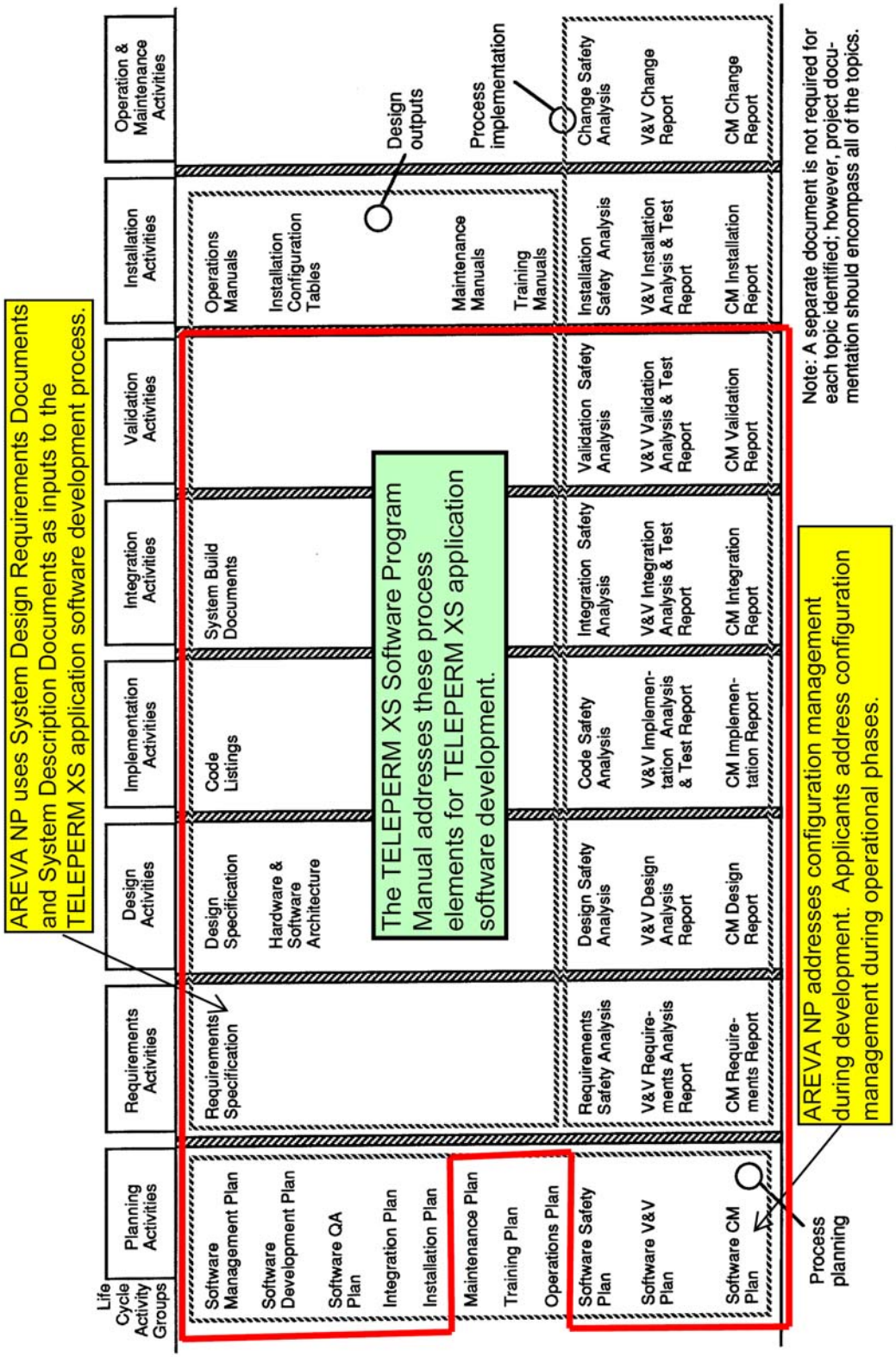


Figure 3-3 - Annotated Version of BTP 7-14 Figure 7-A-1

3.1.5 Relationship to Cyber Security Guidance

The combination of the information in the TELEPERM XS Topical Report on system design features, the information in the Software Program Manual on Application Software development, and the supplemental security-sensitive information provided in a separate letter to NRC (Reference 48) address the cyber security items associated with the Concept Phase through the Test Phase, as described in Regulatory Guide 1.152 (Reference 4). Applicants using the TELEPERM XS technology for safety-related projects will need to address the cyber security items for the Installation, Checkout and Acceptance Testing, Operation, Maintenance, and Retirement Phases. The applicant must address cyber security requirements for TELEPERM XS Gateway communication outside the most secure defensive layer.

3.1.6 Project-Specific Actions Items

Applicants using the TELEPERM XS technology for safety-related projects will need to address the following project-specific action items.

1. The applicant must address the software planning and development activities for the Operations, Maintenance, and Training topics, as described in BTP 7-14.

2. The applicant must address the software planning and development activities for the Installation phase, as described in BTP 7-14.

3. The applicant must address software configuration management, consistent with Regulatory Guide 1.169 (Reference 6), for the Installation, Checkout and Acceptance Testing, Operation, Maintenance, and Retirement Phases, as described in BTP 7-14.

4. The applicant must address the cyber security items, consistent with Regulatory Guide 1.152, for the Installation, Checkout and Acceptance Testing, Operation, Maintenance, and Retirement Phases.

Draft

5. The applicant must address cyber security requirements for TELEPERM XS Gateway communication outside the most secure defensive layer.

3.2 Organization

This Software Program Manual discusses different organizational units and their responsibilities and relationships with regard to software development. The AREVA NP Quality Management Manual (Reference 37) provides the basic organizational structure that governs software quality.

Specific project plans provide project specific organizations. The roles and responsibilities discussed below are given in general terms. The project plans describe any differences from these descriptions. AREVA NP organization charts indicate the reporting relationships.

A separate software integration organization or software installation organization is not necessary for TELEPERM XS projects, since no special software integration effort is required for the TELEPERM XS system software and Application Software. The TELEPERM XS platform is a fully integrated suite of hardware and software designed specifically for nuclear safety applications that has been fully qualified as an integrated platform. Application Software is automatically generated by the TELEPERM XS SPACE tool from Function Diagrams and Network Diagrams. Logical ‘software integration’ occurs at this stage. The project-specific TELEPERM XS system is developed from qualified hardware and software modules using the qualified development tools. Physical software integration occurs during the system test stage, when the Application Software is loaded on the TELEPERM XS processors. The project-specific system test plan covers the approach and activities associated with the software and hardware integration.

The independent verification and validation process is organized under a different reporting chain of command from that of the design functions. Independence is defined as comprising technical, managerial, and financial independence. The Software

Draft

Verification and Validation Plan provides the details of the verification and validation process.

Written procedures and instructions contain the specific responsibilities and authority to ensure software management and control.

According to the requirements of Appendix B of 10 CFR Part 50, the independent QA organization ensures that design and verification and validation activities are performed in accordance with the quality manuals, plans, and procedures which are based on this Software Program Manual.

3.2.1 Roles and Responsibilities

3.2.1.1 Technical Manager

The Technical Manager is responsible for software planning, technical development, integration, installation, and manufacturing testing of the TELEPERM XS Application Software. The Technical Manager ensures that the provisions of this manual, the supporting plans, and the implementing procedures are carried out in each of the phases of software development as specified. The Technical Manager oversees software projects. As such, the Technical Manager is responsible for tracking project indicators and taking corrective action when indications reflect a lack of technical, financial or schedule progress.

The Technical Manager is responsible for the competency and training of the I&C engineering staff and provides objective evidence that each engineer assigned to the development of safety-related TELEPERM XS software has completed the required training requirements. The Technical Manager is responsible for ensuring that all project work is performed in accordance with the applicable QA processes and procedures, as required by the AREVA NP Quality Management Manual.

The Technical Manager is responsible for ensuring design errors are corrected and software changes associated with discrepancies and other anomalies identified by verification and validation activities are implemented. In addition, the Technical

Draft

Manager is responsible for ensuring that any anomalies reported by the customer after the turnover and acceptance of the system are evaluated and resolved.

The Technical Manager is responsible for ensuring that the training supervisor executes the project specific training plans correctly.

Additional responsibilities of the Technical Manager are described in Sections 10.2 and 12.2.

3.2.1.2 Project Manager

The Project Manager ensures that the design, verification and validation, and QA activities on the project are conducted in accordance with this Software Program Manual and the AREVA NP Projects Manual (Reference 38). The Project Manager collects the data to generate the project management indicators for the project.

The Project Manager is responsible for the adequate budgeting, scheduling, and staffing of the project software activities described in this report, including the timely acquisition of independent Verification and Validation resources for the project while maintaining the technical, financial, and organizational independence of the Verification and Validation group. The Project Manager is also responsible for issuing a project plan for the project. Project plans are developed based on specific project tasks and the standard AREVA project management methods. The project plan is updated as required throughout the life of a project.

The Project Manager is responsible for creating and maintaining a project correspondence and filing system and performing periodic configuration status accounting reports on contract deliverables. The Project Manager ensures that any discrepancies and other Open Items that are found during reviews and tests are processed and resolved. The Project Manager is responsible for the identification and subsequent implementation of any project specific training requirements.

Draft

The Project Manager administers the change management program in compliance with the requirements of the Software Configuration Management Plan as well as with the cost control requirements of the project plan.

Additional responsibilities of the Project Manager are described in Sections 3.2, 10.2, and 12.2.

3.2.1.3 Software Supervisor

The Software Supervisor reports to the Technical Manager. The Software Supervisor is responsible for the development of the software for each project. The Software Supervisor assigns a lead software designer for each project. In addition, the Software Supervisor is responsible for the execution of the Software Safety Plan to ensure that no safety hazards exist in the project software. In addition, the Software Supervisor reports the progress of the development to the Project Manager and the Technical Manager.

Additional responsibilities of the Software Supervisor are described in Sections 10.2 and 12.2.

3.2.1.4 Lead Software Designer

The lead software designer reports administratively to the Software Supervisor and functionally to the Project Manager for the project.

The lead software designer leads the Software Engineering group in the preparation and generation of the SRS, related analyses, SDD, and other software products.

The lead software designer resolves any anomalies reported by the customer on systems already turned over to the customer.

3.2.1.5 I&C Engineers

The I&C engineers are responsible for the generation and quality of the software documents as they become the basis of the automatically generated software. The I&C

Draft

engineers report to the Software Supervisor administratively and functionally to the Project Manager through the lead software designer.

The I&C engineers generate the SRS and SDD. Then, the I&C engineers use the TELEPERM XS SPACE engineering tool Application Software. The I&C engineers may also use a NRC-approved tool to perform simulation testing on the software during the software development.

The I&C engineers also install the software prior to the system test phase. AREVA NP can also install the software prior to installation and commissioning, if requested by the customer.

3.2.1.6 Testing

The I&C engineers and hardware engineers install the TELEPERM XS components and hardware accessories and install the specified operating systems and other AREVA NP GmbH platform software. The completed Application Software is then installed onto the TELEPERM XS processors. The Testing group then performs hardware checkouts.

Manufacturing testing is performed by hardware and software engineers. The Testing group may include personnel from the Hardware and Software Engineering groups. The Testing personnel may assist the Verification and Validation group for the preparation of validation test specifications, procedures, and reports or in the performance of test tasks; however, these support personnel shall work under the supervision of the Verification and Validation group. These matrixed personnel bring special skills to supplement the verification and validation activities. These test support personnel may not develop software test documents for design work they prepared.

3.2.1.7 Training Supervisor and Instructors

The training supervisor and instructors are responsible for creating the general and project specific training plans and executing them in a timely manner during the course of the project.

Draft

3.2.1.8 Verification and Validation Group

The Verification and Validation group is responsible for independent verification and validation activities described in the Software Verification and Validation Plan. It may participate, as an observer, in design reviews undertaken by the I&C Systems Software Design group.

The Verification and Validation group is comprised of personnel from AREVA NP, Inc. (NL-A). Additional support can provided by the specialized hardware design group of NL-G, and the specialized Verification and Validation group of Euriware, an AREVA subsidiary based in Europe. The Verification and Validation group is technically, managerially, and financially independent from the design organizations, as required by IEEE Std 1012-1998, as endorsed by Regulatory Guide 1.168. Verification and Validation personnel may not participate in any design preparation activities; however, the Verification and Validation group can perform the Appendix B independent design review for the Software Design group and Hardware Design group, provided the requirements for Appendix B Design Control are satisfied. The Verification and Validation group manager has a dotted line responsibility to the QA group. From a project management perspective, the Verification and Validation group is considered part of the overall project team.

The Verification and Validation group has sufficient resources (budget, staff, etc.) and authority to ensure verification and validation activities are not adversely affected by commercial and schedule pressures.

Additional responsibilities of the Verification and Validation group are described in Sections 11.2 and 12.2.

3.2.1.9 Quality Assurance Group

The QA organization for AREVA NP performs the quality surveillance and audit functions for the software development. The Quality Management Manual describes the QA organization. The corporate QA program maintains the independence of the QA organization. The QA organization reporting chain is through the Vice President

Draft

responsible for U.S. Region Quality, who reports to the President and Chief Executive Officer of AREVA NP Inc.

3.3 TELEPERM XS Application Software Development Controls

The hierarchy of the documents that govern Application Software development is as follows:

- AREVA NP Quality Management Manual (Upper Tier QA Requirements),
- Software Program Manual (Upper Tier Programmatic Requirements),
- Operating Instructions used to implement the various software plans and other topics (detailed generic implementation requirements specified in Operating Instructions), and
- Project-specific plans for the Software Quality Assurance, Software Safety, Software Configuration Management, Software Verification and Validation, or Software Operation and Maintenance (if used for additional project-specific implementation requirements).

Project-specific plans are developed to be sent to a client as a document deliverable that defines the programmatic requirements used for a project activity, when the client requires such documentation. Project-specific plans incorporate the generic program requirements from the corresponding Operating Instruction in effect at that time. Project-specific plans can also augment the procedure requirements established in the generic procedures, as specified in customer requirements or unique system requirements, or take some exception to a provision of the general plan, when justified.

Project plans are developed for each TELEPERM XS project and approved by the Project Manager. The Project plan is a project management document that details topics such as the project controls plan, the overall project quality plan, the project human resources plan, the project communications plan, the project risk management plan, the project procurement/contract management plan, and the project close out plan.

Draft

3.4 Problem Reporting

This section defines the responsibilities and requirements for identifying, processing, and resolving problems and discrepancies regarding the TELEPERM XS system Application Software developed by AREVA NP for use in safety-related I&C applications deployed in the U.S. The problem reporting process handles hardware and software component problems, nonconformances, verification and validation and testing anomalies, reporting of defects and noncompliance in accordance with 10 CFR Part 21 as well as customer suggestions and potential product improvements. The problem reporting process also covers those errors discovered in the system software at the AREVA NP facility and resolved by AREVA NP GmbH.

Employees working on TELEPERM XS projects or using the TELEPERM XS software are responsible for following the methods and principles described in this section. Each employee who identifies a discrepancy, potential for improvement, a nonconformance, or a potential safety concern in connection with a TELEPERM XS software product must ensure that this deficiency or problem is clearly identified, such as by recording the error message, producing screen shot copies, or creating a memory dump.

3.4.1 Corrective Action Program

The AREVA NP Corrective Action Program establishes the process for promptly identifying and correcting conditions adverse to safety and quality in addition to providing the means for customer notification of these conditions. The Corrective Action Program also establishes the means for the identification of and resolution to near miss problems, customer identified problems, and complaints. The condition report process implements a graded approach to managing adverse conditions. Condition report process actions are based on the significance, that is, Levels 1, 2, 3, or 4, associated with the adverse condition. An evaluation is performed and documented in the Corrective Action Program to determine if previous similar projects and customers are affected.

Draft

Items from the Open Items list (discussed in Section 3.4.2) are reviewed for conditions adverse to quality and safety, which are entered into the Corrective Action Program. Additionally, problems identified after delivery (see Section 3.4.6) are entered into the Corrective Action Program. The NRC reporting requirements of 10 CFR Part 21 are then evaluated. If required, a report is made in accordance with the AREVA NP procedures and affected customers are notified.

The Corrective Action Program is implemented as described in Reference 44.

3.4.2 Open Items

Identified issues or Open Items are documented, and the organization responsible for the design evaluates and resolves them. Open Items are collected in a project-specific database as they are identified. The Open Item database tool is selected by the Project Manager in collaboration with the customer. Open Items that involve conditions adverse to quality and safety are entered into the Corrective Action Program.

For each Open Item, a brief description and a reference that describes the origin and the reason for the Open Item is documented in the database.

If an Open Item results in the decision to modify the software and the proposed modification is initially evaluated favorably, it is processed under the change control procedures described in the Software Configuration Management Plan. The project engineer and the Software Supervisor approve the release of each Open Item for implementation.

Software modifications that affect design basis concepts or functional requirements are discussed with the customer. Similarly, if test procedures or hardware are potentially impacted and may have to be changed in connection with the resolution of the Open Item, this is discussed with potentially affected design interface parties.

The Open Item database differentiates between Open Items that result from either testing or being discovered by the design team and Open Items that are found during other review activities. Both the Project Managers of the customer and AREVA NP

Draft

must be involved with the resolution of Open Items that result in the modification of the functional requirements.

The Open Item database differentiates Open Items that are the result of verification and validation activities because these issues are typically discovered later in the design development life cycle; therefore, they pose a greater project management risk. The Verification and Validation engineer verifies the implementation of the Open Item into the project documentation. At the end of each phase of the software development life cycle, the Verification and Validation engineer independently assesses the design process and the project risk associated with software Open Items.

The Open Items database is the primary source of software quality metrics and is reviewed periodically by management to infer the effectiveness of the software QA program. The reviews select specific software quality metrics, such as type of problem, phase and risk of the Open Items, and consider the extent of problem suggested by these metrics.

Although the Open Items database is the tool by which the Open Items are processed and managed, it does not satisfy the record keeping requirements of Appendix B of 10 CFR Part 50. Therefore, individual Open Item forms for project Open Items are stored in the AREVA NP Records Management System at the end of the project as a part of final documentation.

3.4.3 Discrepancies Identified by Design Reviews and Audits

Discrepancies identified during formal reviews and audits are documented in meeting notes, audit reports by the Design Review Board (Reference 40) or another auditing organization. If these discrepancies lead to a potential modification of the current design, an Open Item or Condition Report is released for each discrepancy.

3.4.4 Discrepancies Identified by Testing

Discrepancies identified during testing are first recorded in a test discrepancy log and evaluated with the software engineering group to determine if the problem resolution

Draft

lies in revising the test plan or procedures or if the discrepancy is a software problem that may result in a modification. Problems resolved by revising the test documentation are addressed with in an Open Item or Condition Report and documented in the test log. Problem resolution involving a modification of the software or a revision of the software design documentation is addressed using an Open Item or Condition Report and noted in the test log.

3.4.5 Discrepancies Identified by Verification and Validation

Discrepancies identified by the Verification and Validation group are classified as Open Items. Each Open Item is evaluated for its potential impact on the software system and for the magnitude of the impact. The Open Item is resolved in accordance with the Open Item implementing procedure.

3.4.6 Discrepancies Identified after Release to the Customer

Discrepancies identified after the release to the customer are to be handled in accordance with the Quality Management Manual and the Corrective Action Program. The NRC reporting requirements of 10 CFR Part 21 are evaluated. If required, a report is made in accordance with AREVA NP procedures and affected customers are notified.

AREVA NP and the customer examine the discrepancies jointly for the impact of the discrepancies on the safety functions. The examination determines the rating of the discrepancy:

- No change necessary — The software functions as required by the SRS. A change in the operating procedures may be required, but a modification of the software is not required.
- Change necessary for optimization — A software update can be implemented during the next scheduled software version release.
- Change necessary for fulfilling the required safety function — The responsible design organization will prepare a strategy for immediate action for the release and implementation of a new software update.

Draft

4.0 SOFTWARE DEVELOPMENT PLAN

The Software Development Plan describes the life cycle activities for TELEPERM XS Application Software development.

The TELEPERM XS system is a fully integrated suite of hardware and software designed specifically for nuclear safety applications. The TELEPERM XS system has significant nuclear operating experience. The TELEPERM XS platform has been fully qualified as an integrated platform. The TELEPERM XS system is described in TELEPERM XS Topical Report, which was approved by NRC in May 2000. The TELEPERM XS Application Software development process is tailored to the use of qualified TELEPERM XS hardware and software modules using the SPACE tool.

4.1 Use of TELEPERM XS Technology

The TELEPERM XS System is a fully integrated suite of hardware and software with significant nuclear operating experience. It uses a layered software structure, as shown in Figure 3-2. The TELEPERM XS Application Software development process is founded on several key safety principles. The TELEPERM XS Operating System has a substantial nuclear operating experience base to validate performance and provide opportunities to identify latent errors.

The project-specific Application Software does not have such an experience base. Development of Application Software is always a first-of-a-kind activity. The TELEPERM XS Application Software development process has features that are specifically designed to improve the reliability of the software. For example, the use of a qualified standard TELEPERM XS Function Block (FB) library to generate the Application Software provides a large experience base for the standard modules. The use of the qualified automatic code generators (part of the SPACE tool) eliminates an important human error source by eliminating conventional software development and code generation.

Draft

The SPACE tool eliminates both errors of translation and the introduction of complexity by engineers trying to optimize application coding. Software integration testing (either as an engineering debugging activity or as formal validation testing with a NRC approved test tool) is specifically designed to ensure correct development of I&C functionality in SPACE, which is the first-of-a-kind work in the TELEPERM XS process.

The TELEPERM XS Application Software is developed using object-oriented software development methods in a modular approach that involves both Function Diagrams (FD) and Function Diagram Groups (FDG). FD modules contain code resulting from automatic code generation of FDs being engineered on the SPACE engineering tool. The FD modules implement the code for the engineered application specific I&C functions. The code consists of a sequence of calls to standard FB library functions, as connected for the specific I&C function in the SPACE diagrams. FDG modules also contain code resulting from automatic code generation. The FDG code defines call and data interface to all FD modules running on one TELEPERM XS safety processor. The FDs in a FDG all operate on the same cycle time. A maximum of two FDG modules can be defined for a processor.

The TELEPERM XS platform has been fully qualified as an integrated platform. The overall application independent qualification process is described in Section 2.2 of TELEPERM XS Topical Report. The TELEPERM XS platform qualification process is shown in Figure 3.1. The qualification process is a two-part process: generic system qualification and specific system qualification. The qualification process for Application Software starts with the application-independent (generic) qualification process described in Section 2.1 of the TELEPERM XS Topical Report. The generic qualification process included an integration and system test phase. The specific system used for this generic qualification step is described in detail in Section 3.2.2 of the TELEPERM XS Topical Report.

The generic qualification of the Application Software development process includes work performed by AREVA NP (GmbH) and qualification work performed by independent third-party reviews by independent test institutes: Gesellschaft für

Draft

Anlagen- und Reaktorsicherheit (German Society for Plant Safety and Reactor Safety known as GRS), Institut für Sicherheitstechnologie (Institute for Safety Technology known as ISTec), and Technischer Überwachungs-Verein (German Technical Inspection Agency known as TÜV).

The TELEPERM XS project-specific work (application-dependent) takes credit for all application-independent (generic) qualification activities, as noted on page 2-4 of the TELEPERM XS Topical Report. This Software Program Manual was prepared to describe the project-specific Application Software development process for TELEPERM XS projects in the U.S.

4.2 TELEPERM XS Project-Specific Development Process Overview

The TELEPERM XS Topical Report describes a general framework for the implementation of individual projects using the TELEPERM XS technology in Section 5.1.3. An integral part of that framework is the use of the TELEPERM XS engineering tools set for the development of Application Software. A key feature of the TELEPERM XS tool set is the qualified automatic code generator in the SPACE tool. The development of TELEPERM XS Application Software is predicated on the use of the SPACE tool for code generation.

The TELEPERM XS Topical Report describes the simulator-based testing process for TELEPERM XS Application Software in Section 2.4.3.3.2. The role of the simulator validation tool in the standard AREVA NP engineering process for TELEPERM XS project implementation is shown in TELEPERM XS Topical Report Figure 2.8. The correctness of TELEPERM XS Application Software in the course of specific projects is ensured by software simulation testing (either as an engineering debugging activity or as formal validation testing with a NRC approved test tool).

The software requirements for the Application Software are extracted from the customer specifications and other documents, such as System Requirements and System Design documents, and then documented in the Software Requirements Specification (SRS).

Draft

The SRS becomes the basis for the next detailed level document called the Software Design Description (SDD). In addition to providing a description of how the software will work, the SDD also includes a set of logic diagrams that illustrate software functionality.

These logic diagrams are drawn from a database library of pre-defined logic block pictures contained in a database called FunBase. The building of these logic diagrams is done by I&C engineers. These logic diagrams identify the functions that the Application Software will execute. When the logic diagrams are complete and approved in the SDD, they are manually entered into the SPACE engineering tool. The Application Software to perform the functions is automatically generated by the SPACE tool. This automatic generation of the Application Software code eliminates human error in coding the software. SPACE is a fully qualified tool approved by NRC, which found “that the SPACE tool has the capability and safeguards to ensure that the implementation of the application programs can be successfully accomplished on a plant-specific basis.” Historical information on SPACE tool qualification is provided in Appendix B.

The following development support tools are part of the SPACE Engineering toolset and used for TELEPERM XS Application Software development.

- **dbadmin** is an administration tool that is used to set up, administer, save and identify project databases and is an important tool for configuration management
- **fdprint** is used for printing Function Diagrams for the system to support review meetings and documentation requirements
- **hwparams** is used for listing hardware parameters and / or checking the specification data for compliance with the same rules that are also used by the RTE code generator for consistency check in the course of code generation
- **swparams** is used for listing software parameters and / or checking the specification data for compliance with the same rules that are also used by the FDG code generator for consistency check in the course of code generation

Draft

- **cpuload** is used for load analysis for TELEPERM XS processing units
- **netload** is used for load analysis for TELEPERM XS local area netwrok connections
- **reflist** is an authentication tool used for registering software configurations in directories
- **scanmic** is an authentication tool used for registering the software configuration in loadable MIC files
- **sveload** is a software loading tool used on a portable diagnostics unit, which can be connected to the TELEPERM XS processing module for local software loading. With the aid of the tool, the software module to be loaded is easily selected and then transferred to the processing module where it is programmed into the program memory. The loading procedure is monitored by means of checksum comparison.
- **cmp_code** is used for verification of the scope of a modification in Application Software code generated after implementing a specification change in the SPACE project database.

Application software is developed using the TELEPERM XS Specification and Coding Environment (SPACE) tool. This tool is used to develop Function Diagrams and Network Diagrams. Function Diagrams specify the signal processing requirements for the system. Network Diagrams define the hardware components of the system and their logical interconnections. Software code is automatically generated from the Function Diagrams and Network Diagrams by the SPACE tool. Logical ‘software integration’ occurs at this stage. The project-specific TELEPERM XS system is developed from qualified hardware and software modules using the qualified development tools.

Draft

Physical integration of the Application Software integration with the project-specific TELEPERM XS hardware occurs during the pre-FAT stage, when the project-specific Application Software is loaded on the TELEPERM XS processors. The Application Software Installation Plan is described in Section 7.0. The Application Software Installation Plan is described in Section 7.0. The project-specific FAT Plan covers the approach and activities associated with the Software and Hardware Integration.

A project-specific Software Generation and Download Procedure is issued for each project to control and document the generation of each Application Software release. It is also used to control and document the download of each approved software release to the target system. This project-specific Software Generation and Download Procedure is implemented under a work order for each Application Software Release. The Software Generation and Download Procedure is a configuration item that is governed by the Software Configuration Management Plan.

The software produced by SPACE is checked to ensure it has no syntax errors; however, it may contain design errors made during the logic diagram or functional diagram development. Since human error can occur in the development of the logic diagrams or in the translation of the logic diagrams into functional diagrams in the SPACE tool, the software is validated with software integration and system testing, either using a NRC approved simulation test tool or in the factory acceptance test field. This testing is designed to ensure the SPACE generated software will function properly. If a functional error is detected, the functional diagrams are examined to determine the source of the error. The error is corrected using the appropriate design control and configuration management processes, and the testing is repeated. This process is continued until all the functions are correct. The Application Software validation test plans, procedures, and results are prepared by Independent Verification and Validation Group.

4.3 TELEPERM XS Application Software Life Cycle

The TELEPERM XS projects at AREVA NP proceed in the following phases:

Draft

1. Elaboration of the Quotation
2. Project Start-Up
3. Basic Design
4. Detailed Design
5. Manufacturing
6. System Integration and Testing
7. Installation and Commissioning
8. Final Documentation

The development of safety Application Software also progresses according to a formally defined life cycle. The software development life cycle is a defined sequence of activities with specified inputs and outputs that are to be performed during a TELEPERM XS Application Software project. This plan maps to the set of life cycle activities provided in Institute of Electrical and Electronic Engineers (IEEE) Std 1074-1995 (Reference 29), as endorsed by Regulatory Guide 1.173 (Reference 10). The mapping of the AREVA NP software life cycle activities to IEEE Std 1074 software life cycle activities is shown in Appendix A. The mapping is done to the 1995 version to show alignment to Regulatory Guide 1.173. The 1997 version of the standard was included in the mapping since it has a more logical flow of the life cycle activities (Reference 30).

The two versions of IEEE Std 1074 provide Software Life Cycle Models for the development process. The AREVA NP Software Life Cycle was mapped to the suggested software life cycle models given in the two versions of IEEE Std 1074. This mapping was done to identify any activities had been omitted from the AREVA NP Software Life Cycle Model. The alignment for the AREVA NP activities and the two versions is demonstrated in the mapping.

Application Software development activities begin in the basic design phase and are completed during the final documentation phase with the submittal of the final documentation

Draft

The software life cycle activities for these projects fit into the following phases.

- Basic Design
  - System Requirements
  - System Design
  - Software Requirements
  - Initiate Software Requirements Traceability
  - Reports for Verification and Validation Activities (i.e., Acquisition Support, Planning, Concept, and Requirements)
- Detailed Design
  - Software Design
  - Automatic Code Generation
  - Application Software Integration Validation Test Planning (using NRC approved simulation test tool)
  - Application Software Integration Validation Test Execution (using NRC approved simulation test tool)
  - Application Software Integration Validation Test Reporting (using NRC approved simulation test tool)
  - Software Safety Analyses
  - Continue Software Requirements Traceability
  - Reports for Verification and Validation Activities (i.e., Design and Implementation)
- Software Integration and System Testing – including FAT
  - Integration of Hardware and Software
  - Software Integration, System and Acceptance Validation Test Planning
  - Software Integration, System and Acceptance Validation Test Execution

Draft

- Software Integration, System and Acceptance Validation Test Reporting
- Continue Software Requirements Traceability
- Reports for Verification and Validation Activities (i.e., Test)

- Installation and Commissioning
  - Installation and Commissioning Test Planning
  - Installation and Commissioning Test Execution
  - Installation and Commissioning Test Reporting
  - Reports for Verification and Validation Activities (i.e., Installation and Checkout)
- Final Documentation
  - Generation of final documentation before system is turned over to the customer.

The activities of the software plans listed in Section 1.0 fit into this project software life cycle structure. This Software Program Manual discusses each plan from the perspective of the type of activities that are covered by the plan. The plans fit together in the project structure to ensure the end product complies with the governing regulations and standards.

The Software Quality Assurance Plan describes the QA activities in the development of the Application Software. It covers in detail how those activities are accomplished in a quality manner.

The Software Safety Plan describes the activities during the development of the Application Software that ensure that the safety function of the software meets the design goals. Adherence to the Software Safety Plan provides assurance that the Application Software will perform its safety function during system operation.

Draft

The Software Verification and Validation Plan describes the method of assessing the quality of the software. Adherence to the Software Verification and Validation Plan ensures the verification of an accurate translation from each software development process and the validation that the software product fulfills the requirements for which it was developed.

The Software Configuration Management Plan describes the procedures necessary to maintain the software in an identifiable state during the development process. If additional development work is requested by a customer during the operations phase, the Software Configuration Management Plan ensures the installed software to be modified is the appropriate approved version before work starts and controls the version as work progresses.

The Software Operation and Maintenance Plan describes the activities necessary to maintain the software, respond to new or revised requirements, and adapt the software to changes in operating environments. The recommendation is made for a Software Operation and Maintenance Plan to be administered by the customer.

The Software Training Plan describes the activities necessary to produce a project training plan that meets the customer's needs and conforms to applicable regulatory standards.

This Software Program Manual and the subordinate plans are written and maintained following the AREVA NP Records Management Program Manual (Reference 33).

The Software Development Documentation section, Section 4.5 of this Software Program Manual, describes the required documents.

The problem reporting and corrective action section, Section 3.4 of the Software Program Manual, describes procedures to ensure that software errors and failures are promptly acted upon in a uniform manner.

Draft

4.4 Software Classification

The TELEPERM XS Application Software design reflects the requirements specified in the SRS, which are derived from the system requirements and system design activities. The Application Software performs the respective functions, establishes communication channels to other modules, and operates the input/output (I/O) modules. Because a fault in this software could prevent the TELEPERM XS modules from performing the nuclear safety functions, Application Software is safety critical. Therefore, the TELEPERM XS system is classified as Class 1E for nuclear safety systems as defined by IEEE Std 603-1991 (Reference 18), and the Application Software is classified as Software Integrity Level (SIL)-4 as defined by IEEE Std 1012-1998 (Reference 25). All software on the safety processors and the Monitor and Service Interface (i.e., the processors performing nuclear safety functions) is classified as SIL-4.

Other project-specific software that does not perform design basis accident mitigation functions directly may be classified with a lower SIL classification that is appropriate to the relative importance to safety, such as the software running on the TELEPERM XS Gateways, on the Graphic Service Monitor, and certain other portions of the software supporting functions. These software elements do not run on the safety processors computers and do not perform any safety functions. They can be classified at lower SIL levels than the safety-related Application Software running on the safety processors because they are not directly a part of the safety function. A criticality analysis determines the appropriate SIL classifications and assigns the classifications following the guidance of IEEE Std 1012-1998, as endorsed by Regulatory Guide 1.168 (Reference 5).

4.5 Software Development Documentation

The software development life cycle process at AREVA NP has been mapped to the software life cycle model presented in IEEE Std 1074-1995. This section presents the documents that are used during that process and the associated requirements.

Draft

4.5.1 System Design Requirements Document

A SDRD is issued to document all requirements that are applicable to the design of the TELEPERM XS safety system, including functional and other requirements. Functional logic diagrams may also be used to define high level functional requirements of the system. The SDRD may be jointly developed by AREVA NP and an external customer for upgrade projects, or be developed by AREVA NP for new plants. The key system information required for the Application Software development include requirements for:

- I&C functionally (for protection and control),
- Alarm and annunciation,
- Safety-related to non-safety related interfaces,
- Access control (including cyber security),
- Reliability, maintainability, testability, and fault accommodation, and
- Data export.

For upgrade projects, the SDRD may also derive requirements from an external customer specification.

4.5.2 System Design Description

The SysDD describes the functional design of the I&C system. This includes the system architecture and allocation of functional requirements within the architecture. Functional logic diagrams may also be produced to illustrate high level functional behavior of the system.

4.5.3 Software Requirements Specification

The TELEPERM XS Topical Report and the associated safety evaluation report issued by the NRC document a number of general TELEPERM XS software design constraints and performance requirements. Those requirements allocated to software based system elements during the system design activity, along with the general TELEPERM

Draft

XS software design constraints, serve as the basis for the software requirements stated in the SRS.

Other potential sources of added requirements include the results of the system safety analyses, human factors engineering reviews, and any other applicable regulatory requirements and commitments.

The SRS forms an essential part of the record of the design of safety system software. Software requirements are associated with system requirements that are allocated to software subsystems and serve as the design bases for the safety software to be developed. Therefore, the SRS is a crucial design input to the remainder of the software development process.

The SRS conforms to the guidance of IEEE Std 830-1993, as endorsed by Regulatory Guide 1.172. The specific requirements are organized by I&C function (e.g., protection or control), and each function is subdivided into sections for functional, I/O, manual actuation and display requirements, and maintenance and testing requirements. Graphical representations of software requirements in the form of logic diagrams may be used. The FunBase tool may be used for this purpose.

Software requirements for fault tolerance and failure modes are derived from the consideration of system level safety analyses and the FMEA, and are specified for each operating mode.

Software behavior in the presence of unexpected, incorrect, anomalous, and improper input, hardware behavior, or software behavior is fully specified.

Software requirements for responding to both hardware and software failures are provided, including the requirements for the analysis of and recovery from computer system failures. Requirements for online or in-service testing and diagnostics are specified.

Draft

Software actions that are required to detect, prevent, or mitigate security threats are specified, including access control restrictions, such as modification of instrument calibration data being protected by a password system.

For safety system software, software requirements important to safety are identified as such in the SRS. Software requirements which are classified as non-safety are linked to either site and equipment variations or to specific plant design bases and regulatory provisions in the SRS. Regardless of whether it performs a safety function, software running on a processor with safety software is classified as safety-related (SIL-4). Unnecessary requirements are not imposed on safety system software.

4.5.4 Software Design Description

The SDD implements the requirements from the SRS into a logic diagram-based representation of the software design. The logic diagrams show the inputs, how those inputs are manipulated, and the resulting outputs. The SDD logic diagrams can be drawn directly using the SPACE tool, or using the FunBase tool. If the FunBase tool is used, these diagrams are redrawn in the SPACE tool to generate the code.

4.5.5 Application Software Requirements Traceability Matrix

The Application Software Requirements Traceability Matrix is used to ensure that the requirements have been successfully implemented in the design and tested in the testing program.

The Software Requirements Traceability Matrix supports traceability analysis. Each identifiable requirement in the SRS is traceable backwards to the system requirements and either the design bases or regulatory requirements that it satisfies. Each identifiable requirement is written so that it is also traceable forward to subsequent design outputs, that is, from the SRS to the SDD and from the SDD to the test plan and test procedures.

The Software Requirements Traceability Matrix is implemented in a requirements database and is updated continuously throughout the software development process.

Draft

Both periodically and at the end of each phase of the software development life cycle, exporting selected views from the database of the Software Requirements Traceability Matrix creates a report. The minimum view is a screenshot of the SRS requirements showing “traced-from” and “traced-to” links. The exported views are stored in the AREVA NP Records Management System.

4.5.6 Test Plan

Software testing follows written test plans that conform to the guidance of IEEE Std 829-1983, as endorsed by Regulatory Guide 1.170. The test plan prescribes the scope and approach of the testing activities. It identifies the items to be tested, the features to be tested, the testing tasks to be performed, the organization responsible for the testing, and the risks associated with the plan.

The test plan is prepared during the detailed design phase of the software development life cycle. Prior to the commencement of formal testing, the test plan is reviewed for completeness, accuracy, and consistency.

Application Software validation testing is planned and executed in accordance with procedures that conform to the applicable guidance of IEEE Std 1008-1987, as endorsed by Regulatory Guide 1.171 (Reference 8).

The test plan addresses the objectives of test coverage. The aspects of test coverage important for the testing of safety system software include the coverage of requirements, traceability of requirements, and coverage of the internal structure of the code.

4.5.7 Test Documentation

The documentation used to support software testing includes the information necessary to meet regulatory requirements as applied to software test documentation. At a minimum, this information includes:

- Qualifications, duties, responsibilities, and skills required of the persons and organizations assigned to testing activities

Draft

- Environmental conditions and special controls, equipment, tools, and instrumentation needed for the accomplishment of testing
- Test instructions and procedures incorporating the requirements and acceptance limits in applicable design documents
- Test prerequisites and the criteria for meeting them
- Test items and the approach taken by the testing program
- Test logs, test data, and test results
- Acceptance criteria
- Test records indicating the date of the test, the identity of the tester, the type of observation, the results and acceptability, the action taken in connection with any deficiencies, and the signatures of the testers

A handwritten logbook may be used for the test logs kept during testing activities. Alternatively, a voice activated recorder may be used to allow the collection of some log data without having to delay testing to make manual log entries. Regardless of the data capture method, the data is stored as a quality record in accordance with the AREVA NP Quality Management Manual procedures.

4.6 Security and Disaster Recovery

As recommended by Regulatory Guide 1.152, software security activities are specified for each phase of the Application Software life cycle. The combination of the information in the TELEPERM XS Topical Report on system design features and the information in this section address the cyber security items associated with the Concept Phase through the Test Phase, as described in Regulatory Guide 1.152.

4.6.1 Cyber Security Design Features

The TELEPERM XS system was designed to address the nuclear safety aspects of interference-free communication and safety to non-safety system isolations. These TELEPERM XS design features, which provide control of access, communication

Draft

independence, safety to non-safety system isolation, and interference-free communication, also address the cyber security attributes of confidentiality, integrity, and availability. The key cyber security features of the standard TELEPERM XS platform are contained within the design of the processors and operating system and associated access control features.

The standard TELEPERM XS platform (hardware and operating system) was designed several years prior to the issuance of Regulatory Guide 1.152 (Reference 4). As such, some elements of the generic system qualification are not explicitly addressed as cyber security activities in the TELEPERM XS Topical Report and the associated NRC safety evaluation report. The TELEPERM XS software development process controls related to cyber security are briefly mentioned in the TELEPERM XS Topical Report but not mentioned in the NRC Safety Evaluation Report. The TELEPERM XS system supports bidirectional communication with non-safety systems within the most secure defensive layer used for safety-related and other critical control systems. TELEPERM XS Gateway communication outside this defensive layer will need additional protection (e.g., firewalls of unidirectional control), which are determined for each project in coordination with customer cyber security controls. Additional information on the TELEPERM XS design features that provide cyber security protection was provided to NRC in a separate security-sensitive submittal (Reference 48).

The safety system security requirements identified in the SRS are translated into specific design configuration items in the SDD during the detailed design phase. The safety system security design configuration items address the control over physical and logical access to the system functions, the use of safety system services, and data communication with other systems. The guidance of Regulatory Guide 1.152 is implemented as follows:

- Items C.2.1 through C.2.4 of Regulatory Guide 1.152 are used as design inputs to ensure conformance
- Item C.2.5 is used as a design input into the test plan and test procedures to ensure that security features are properly tested

Draft

4.6.2 Cyber Security Administrative Controls

Information on the procedure controls currently in effect for the development of TELEPERM XS software (Operating System Software, Function Block Library, and Application Software) that provide a secure software development infrastructure was provided in the supplemental security-sensitive information provided in Reference 48.

AREVA NP will have implementing procedures to address disaster recovery for each project.

Items C.2.6 through C.2.9 of Regulatory Guide specify guidance to be addressed by the customer and do not apply to AREVA NP. AREVA NP Inc. will support the customer's efforts on the development of security procedures and assessments. Additional requirements for AREVA NP Inc. may be specified by contract and customer programs, policies, and procedures. Applicants using the TELEPERM XS technology for safety-related projects will need to address the cyber security items for the Installation, Checkout and Acceptance Testing, Operation, Maintenance, and Retirement Phases.

4.7 Documentation Standards

AREVA NP has documentation standards that apply to the documents produced during the software development process.

4.7.1 Naming Conventions

The identification (ID) coding standards for the SRS are based on the description of functions in the SDRD. ID codes for signals, transmitters, and actuators are described in interface lists and the project-specific ID coding concept. Human factors engineering reviews are performed to ensure consistency with operations and maintenance procedures and TELEPERM XS user manual documentation.

4.7.2 Coding Standards

The SDD is documented to enable the unique identification of ID codes for transmitters, signals, actuators, and annunciators. The safety functions, conditioning, and online validation of input signals, actuation, and annunciation is documented in uniquely

Draft

identified modules. Each module is described separately, including the interfaces and connections to other modules; therefore, each module can be tested and replaced independently from other modules. Part of the design verification is the verification of the SDD conformance to the coding standards.

4.7.3 Logic Structure Standards

The logic structure of the SDD, in general, and the module descriptions follow the structure of the TELEPERM XS FunBase database. The FunBase tool offers a graphical user-interface that allows the functions to be designed according to the specification in the SRS. Part of the verification and validation activities is the independent verification of the consistency between the logic structure and the functional requirements as described in Section 11.0.

4.7.4 Function Diagram Standards

The SPACE function diagrams have the same logical structure as the diagrams shown in the SDD. They contain information about setpoints and time-delays as well as interface information, as described in the SDD. In practice, SPACE function diagrams are specified for each function. ID codes correlate SDD modules and the function diagrams. The SPACE tool creates a project database that stores the information about the safety functions, the function-specific setpoints, inputs, and outputs. The resulting function diagrams are a graphical interpretation of the safety system. The SPACE database forms the basis for the automatic code-generation. The verification and validation activities include the independent verification that the SDD conforms to the functional requirements and design constraints.

4.7.5 Code Configuration

The TELEPERM XS code generators create the source code for the safety function Application Ssoftware. The source files are generated in the programming language ANSI-C. The created source code files are not to be modified manually, and they are subject to the rules described in the Software Configuration Management Plan.

Draft

4.8 User Manuals

AREVA NP Inc. prepares user and maintenance manuals following the recommendations of IEEE Std 1063-2001 (Reference 28). Standardized nomenclature and abbreviations are used for equipment in text, labels, and drawings.

Human factors engineering is considered in the development of documentation which accompanies each piece of equipment, such as user manuals, handbooks, or parts lists. Engineering drawings provide additional perspective for the users. Some considerations include thoroughness, technical accuracy, format of the documentation, quality of illustrations, and the reading level and technical sophistication required to understand the documentation.

Draft

5.0 SOFTWARE QUALITY ASSURANCE PLAN

5.1 Purpose

The Software Quality Assurance Plan describes the necessary processes that ensure that the software attains a level of quality commensurate with its importance to safety function.

The Software Quality Assurance Plan describes the tools and methodology to be followed during the development and maintenance of Application Software developed for TELEPERM XS projects. The Software Quality Assurance Plan fulfills the requirements for a software QA plan and conforms to IEEE Std 730-2002 (Reference 20). Some of the activities described in the Software Quality Assurance Plan are performed by the independent AREVA NP QA organization. The governing document for activities performed by the QA organization (i.e., reviews, surveillances, and audits) is the AREVA NP Quality Management Manual. This information is not duplicated in the Software Quality Assurance Plan; instead, it is simply referenced.

The Software Quality Assurance Plan describes the methodology by which software and documentation is managed throughout the software development life cycle at AREVA NP. The following software elements are produced in the QA process:

- Test plans, cases, procedures or reports
- Review and audit results
- Problem reports and corrective action documentation
- Software configuration management plans
- Software verification and validation plans

5.2 Management

The Technical Manager manages the Software Quality Assurance Plan. The QA group verifies that the implementation of QA requirements is in accordance with the Quality

Draft

Management Manual. The Technical Manager ensures that software and associated documentation has been developed in accordance with the Software Quality Assurance Plan, which includes ensuring that the manufacturing and equipment qualification requirements have been followed. The Verification and Validation group manager ensures that the Software Verification and Validation Plan and test documentation requirements have has been followed.

5.3 Documentation

The Software Quality Assurance Plan specifies the minimum documentation required for each project and the governing plans for the organization. The documentation required for each project is produced and independently reviewed and takes the quality factors listed in Section B.3.3 of BTP 7-14 into consideration.

The additional plans required by the Software Quality Assurance Plan are identified below. Project documentation used as design input or delivered to the customer as design output is stored in the AREVA NP Records Management System. Project records arising from QA inspections and audits are stored in the AREVA NP Records Management System.

The following additional plans are linked to the Software Quality Assurance Plan:

- Software Safety Plan
- Software Configuration Management Plan
- Software Verification and Validation Plan
- Software Operations and Maintenance Plan

5.4 Software Reviews

Software reviews are conducted in accordance with IEEE Std 730-2002 and IEEE Std 1028-1997 (Reference 26), as endorsed by Regulatory Guide 1.168. Reviews take place throughout the software lifecycle and verify that the software products of each phase are correct with respect to the phase inputs and outputs. Review activities are

Draft

detailed in the individual project schedules. A minimum set of reviews are conducted as described in Sections 5.4.1 – 5.4.7.

5.4.1 Software Requirements Review

The Software Requirements Review takes place during the Basic Design Phase after the SRS is completed. The Software Requirements Review shall verify that the SRS was created in accordance with the standards listed in Section 4.5.3 and therefore is unambiguous, complete, verifiable, consistent, modifiable, traceable, and usable during operation and maintenance. Compatibility of interfaces, adequacy of the human-machine interface, and the correctness of logical descriptions shall also be checked. A Software Requirements Review Report is prepared by the Verification and Validation Group and contains information about review activities, review participants, review results, and all identified discrepancies. This review is credited for the Software Specification Review.

5.4.2 Preliminary Design Review

The Preliminary Design Review takes place at the end of the Basic Design Phase and is conducted by a Design Review Board (Reference 40). A Preliminary Design Review shall be performed to verify the technical adequacy of the basic design (system and software architecture), check the compatibility of the functional and performance requirements for the system, and verify whether the interfaces between the software and hardware are consistent. This review is credited for the Architecture Design Review. (For a small project, this review may not be required. This exception shall be taken in the specific project plan.)

5.4.3 Detailed Design Review

A Detailed Design Review shall be performed to verify that the detailed design (i.e., the Software Design Description and the SPACE function diagrams) satisfies the requirements of the SRS and satisfies all functions specified in the SDRD. It also shall assure that the described interface is completely implemented, and requirements for testing are defined. The design review shall verify that the software design is traceable to the requirements. This review is performed by the Software Design Group.

Draft

5.4.4 Software Verification and Validation Plan Review

The Software Verification and Validation Plan Review summarizes all results of the execution of the Software Verification and Validation Plan. It shall list all deficiencies found and provide the results of reviews, audits and tests. The result of the Software Verification and Validation Plan Review shall be the statement that the software can or cannot be released for operational use.

5.4.5 Managerial Reviews

Managerial Reviews may be held periodically by the Project Manager and/or other AREVA NP Senior Managers throughout the design and test processes to assess the execution of the quality requirements in the contract specifications. If the review results in recommended changes or findings, the problem reporting process outlined in Section 3.4 shall be followed.

5.4.6 Software Configuration Management Plan Review

The Software Configuration Management Plan Review is held prior to the start of the design phase to evaluate the adequacy and completeness of the configuration management methods defined in the software configuration management plan.

5.4.7 Post-implementation Review

The Post-implementation review is held at the conclusion of the project to assess the development activities implemented on the project and to provide recommendations for appropriate actions.

5.5 Software Audits

Software Audits are conducted through-out the software life cycle and provide an independent evaluation of conformance of the software products and processes to applicable regulations, standards, and procedures. These audits are the responsibility of the AREVA NP QA organization and are performed using an independent qualified lead auditor and may include technical resources such as Verification and Validation

Draft

personnel, as necessary. The Software Audits conform to the guidance of IEEE Std 1028-1997, as endorsed by Regulatory Guide 1.168.

5.5.1 In-Process Audits

The reviews, inspections, and requirements tracing activities described in Section 11.2.7 are credited for satisfying the in-process audit requirements of IEEE Std 730-2002. These independent verification inspections and reviews are performed by the Verification and Validation Group on software development products, including the SDRD, SRS, SDD, test plans, specifications, cases, procedures, and results. Also included are code reviews for software not generated by SPACE. Deviations or discrepancies are recorded as Open Items. These reviews and inspections are held during the design phase and verify the consistency of the design.

5.5.2 Physical Audits

The Physical Audit is held prior to software release and verifies internal consistency of the software and its documentation, and their readiness for release. As part of the physical audit, current versions of all programs loaded on the hardware, and all design and testing tools shall be audited and compared against the version in the software library, and against the configuration status reports issued under the Software Configuration Management Plan. Identified discrepancies are reported in the form of Open Items and are subject to the problem reporting process outlined in Section 3.4.

5.5.3 Functional Audits

The Functional Audit is held prior to software delivery to verify that all requirements specified in the SRS have been met. The audit shall verify that acceptance test data is complete, accurate and addresses all areas specified in plans, specifications, and procedures.

5.5.4 Software Process Audits

Software Process audits are conducted annually and are held to verify compliance with applicable software development requirements.

Draft

5.6 Testing

FAT determines whether or not a system satisfies its acceptance criteria and enables the customer to determine whether or not to accept the system. During the testing, the code is exercised via the execution of test cases to ensure that the system adheres to the requirements and that the software produces the correct or expected results for each specified combination of inputs.

5.6.1 Test Planning

Testing activities define the systematic, sequential progression of operations and account for the preparation and control of procedures and work instructions. Testing activities conform to the guidance of IEEE Std 829-1983 (Reference 22), as endorsed by Regulatory Guide 1.170 (Reference 7).

Test plans, test specifications including test procedures, and test report documentation are the three types of documents used. These documents are project specific. Each document is independently reviewed before approval.

Test plans cover the subjects recommended in IEEE Std 829-1983.

5.6.2 Test Specifications

Specifications are established for the processing, analysis, and evaluation of inspection and test data and for reduction of test data for prompt evaluation against acceptance criteria. Testing can be performed at various module and unit levels.

The specifications provide acceptance criteria for immediate evaluation to determine the validity of the inspection and test results and the appropriateness of continuing the test or inspection.

Tests demonstrate the required independence or dependence of subsystems. Tests also demonstrate that the operation gives the expected or desired result.

Draft

5.6.3 Test Reporting

Test results are documented in a suitable report, which includes the items tested, the procedures, any unanticipated conditions, the identity of inspector or tester, and the completion date.

Test and inspection records identify any required additional inspection or tests and any changes to the procedures.

The discrepancies that are found during the testing are reported in the form of Open Items and are handled through the problem reporting process outlined in Section 5.8.

5.7 Standards

Section 15.0 lists the regulations, regulatory guidance, and industry standards that are used in the creation and administration of the software program.

5.8 Problem Reporting and Corrective Action

The Open Items process is an administrative system that initially deals with issues that arise during the course of a project. It tracks and closes design and programmatic issues. Section 3.4 contains further discussion of the details of the Open Items system.

The AREVA NP formal problem reporting and Corrective Action Program (Reference 44) documents and processes the anomalies, discrepancies, and Open Items that represent conditions adverse to quality found during the development of the Application Software for the TELEPERM XS systems.

5.9 Tools, Methodologies, and Metrics

5.9.1 Tools and Methodologies

This section identifies the software tools that support the design and verification processes.

Draft

5.9.1.1 Methodology for Generating the Software Requirements Specification

The Software Design group ensures that the SRS conforms to the guidance in IEEE Std 830-1993 (Reference 23), as endorsed by Regulatory Guide 1.172 (Reference 9). The Software Design group ensures that the software design incorporates the customer specification requirements, functional requirements, and software requirements. The Verification and Validation group traces the customer requirements from the SDRD through the SysDD into the SRS. Alternately, the software engineer can create the Software Requirements Traceability Matrix and the Verification and Validation engineer performs an independent traceability analysis to verify that the software requirements in the SRS and derived from the software safety analyses.

5.9.1.2 Tools for Generating the Software Design Description

The Software Design group uses the TELEPERM XS tool, FunBase, to create the SDD. FunBase is a database management tool that is designed to facilitate the organization of the Application Software functions and the respective internal and external I/O signals. FunBase controls the assignment of module and signal naming so that each entity in the software is uniquely and unambiguously named. The Verification and Validation group traces software requirements from the SRS into the SDD. Alternately, the software engineer can create the Software Requirements Traceability Matrix and the Verification and Validation engineer performs an independent traceability analysis to verify that the software requirements in the SRS and derived from the software safety analyses.

5.9.1.3 Tools for the Specification and the Generation of the Application Software

The SPACE engineering system contains the tools for converting function diagrams into software code and includes the source code-generators, such as function diagram group module and run time environment, and the software for compiling, linking and locating, such as make command. These tools are part of the qualified TELEPERM XS software package.

Draft

5.9.1.4 Tools for the Requirements Traceability Matrix

Either a worksheet matrix or a formal requirements management tool can be used for requirements tracing.

5.9.1.5 Tools for Verification and Validation

The following tools are used for verification and validation activities:

- The Software Requirements Traceability Matrix traces requirements through the various design documents (SDRD, SysDD, SRS, and SDD, including the SPACE diagrams (used to generate the code).
- The TELEPERM XS software tools package (such as SPACE, which includes reflist, hardware parameters, software parameters, cpuload, and netload) verifies and documents the software code.
- NRC approved tools for Application Software validation testing.
- The test environment of the field test equipment, including the TELEPERM XS test machine, tests the software implementation onto the system.

5.9.2 Metrics

Software quality metrics are used throughout the software life cycle to assess the effectiveness of the software QA program. Software and design errors are recorded as Open Items during each phase of development. These items are trended to determine the progress in eliminating the errors present in the software and design.

5.10 Media Control

The following subsections define the media for storing each deliverable work product and associated documentation and describe the safeguards used to protect the storage media from unauthorized access and inadvertent damage.

Draft

5.10.1 Media Control for Application Software

The SPACE project database is stored on a network server. Only AREVA NP employees or its contractors can access the server, and only the employees involved in the specification process can access the project database.

The Software Supervisor is responsible for the organization of data backup procedures, schedules, and the storage of data saved on removable media.

5.10.2 Code Control of TELEPERM XS System Software

The TELEPERM XS system software is a ready-made software product, which is not to be modified during any phase of the software life cycle. Before each installation, the identity of the system software must be verified.

5.11 Supplier Control

5.11.1 Software Development by Third Parties

In addition to the system software developed by AREVA NP GmbH, additional third party non-safety software that is rated SIL-3 or SIL-2 may have to be developed to support the projects. Contracts with any third party supplier of software include the provisions of the Software Quality Assurance Plan.

5.11.2 Existing Software Developed by Third Parties

AREVA NP GmbH developed the TELEPERM XS system software and implemented an approved software QA program for the life cycle of the TELEPERM XS software. AREVA NP GmbH is an approved supplier for AREVA NP.

Software in the TELEPERM XS system software package is uniquely identified and is subjected to an incoming inspection and is baselined for configuration control.

No other safety-related software (SIL-4) is required to be procured during the software life cycle of TELEPERM XS projects at AREVA NP.

Draft

5.12 Records Collection, Maintenance, and Retention

Record copies of completed procedures, reports, personnel qualification records, measurement and test equipment calibration records, inspection and examination records, and data analysis and evaluations are prepared.

In accordance with AREVA NP procedures, documents produced during the project are stored in the AREVA NP Records Management System. This includes, but is not limited to, the following documents:

- Software Requirements Specification
- Software Design Description
- Code Documents, such as SPACE listings, code configurations, and the list of changeable parameters
- Verification and Validation Reports
- Test specifications, procedures, and test-results of the software integration validation tests
- Test specifications, procedures, and test-results of the system validation tests (including FAT)
- Software Requirements Traceability Matrix

5.13 Training

In accordance with AREVA NP procedures, Design and Verification and Validation personnel are trained on the provisions of this Software Program Manual.

5.14 Risk Management

The Project Manager identifies and assesses the technical, schedule, and regulatory risks of the project. The assessment is published to an audience specified in the project plan.

Draft

5.14.1 Risk Management Process

The Project Manager uses AREVA NP's standardized project management risk assessment tools to assess project risks. The Project Management Guideline on risk management (Reference 39) documents the method and procedure to identify, assess, monitor, and control areas of risk that arise during the software development project. This methodology is used to identify, assess, monitor, and control areas of risk that arise during the software development project. The risk management process is comprised of the following steps; risk identification, analysis and prioritization, response development, and risk monitoring and control. The methodology utilizes a process to rate the complexity and risks of projects to optimize project planning and execution. In the course of project execution, the project risks are monitored, and the original rating is reviewed to determine if the rating needs to be modified.

The risk assessment process considers a wide number of internal and external factors that can affect project risk, including technical factors. The risk assessment takes into accounts both the use of the TELEPERM XS technology described in the TELEPERM XS Topical Report and the structured software development process described in the Software Program Manual, both of which are designed to deliver high quality digital safety systems.

The risk assessment process also considers project-specific projects risk based on the nature of the specific project. These project-specific factors include, but are not limited to, the development environment, program constraints, integration with other plant systems, project size and complexity, and program and organizational interfaces.

The AREVA NP risk management process conforms to the guidance in IEEE Std 7-4.3.2-2003 clause 5.3.6 (Reference 14), as endorsed by Regulatory Guide 1.152.

5.14.2 Independent Risk Analysis

The Verification and Validation group performs an independent risk analysis at each phase the verification and validation activities, as required by the Software Verification and Validation Plan. The Verification and Validation group provides their

Draft

recommendation regarding continuation into the next phase of the life cycle based on consideration of any Open Items or design issues. The Verification and Validation group also recommends appropriate risk mitigation steps. The results of the risk analysis and any mitigation recommendations are documented in the Verification and Validation Report for each activity phase.

5.14.3 Standard TELEPERM XS Risk Mitigation Measures

Technical risks associated with TELEPERM XS technology are addressed for four areas: software and hardware integration, communication independence, first-of-a-kind engineering work, and software common mode failure.

The TELEPERM XS technology is a mature and fully integrated nuclear safety system. The TELEPERM XS hardware is fully qualified safety-related equipment. The TELEPERM XS operating system software and Function Block library are developed and maintained using the process described in the TELEPERM XS Topical Report. The Application Software is generated by the SPACE tool. The generic TELEPERM XS qualification process removes risks associated with integration of TELEPERM XS software and hardware. Design features of the TELEPERM XS system that address communication independence are addressed in the TELEPERM XS Topical Report.

The risks associated with first-of-a-kind engineering work are minimized through the use of qualified software development tools and structured engineering analyses. The use of the object-oriented automated code generation tool (SPACE) supports the development of high quality software with a less complex process, which minimizes the potential for human error and reduces the inherent risk in the development of the Application Software. Software integration and system testing is used for Application Software validation testing to detect software errors that would prevent the Application Software from fulfilling its safety function. The use of the standard TELEPERM XS tools supports the development of high quality software.

This Software Program Manual describes the program measures incorporated at AREVA NP to ensure that the TELEPERM XS Application Software attains a level of

Draft

quality commensurate with its importance to safety functions. This Software Program Manual uses the following plans to support the development of high quality Application Software and minimize development risks: Software Quality Assurance Plan, Software Safety Plan, Software Verification and Validation Plan, Software Configuration Management Plan, and Software Operations and Maintenance Plan.

An inherent risk of utilizing digital control systems in safety-related applications is the possibility of software common mode failures which could defeat hardware redundancy. A Defense-in-Depth and Diversity Analysis is performed to ensure that adequate defense-in-depth is provided in the design. The Defense-in-Depth and Diversity Analysis addresses residual software risks by addressing mitigation of assumed software common mode failures.

Draft

6.0 SOFTWARE INTEGRATION PLAN

The Software Integration Plan describes the Application Software integration process and the System hardware/software integration process for TELEPERM XS projects.

The TELEPERM XS system is a fully integrated suite of hardware and software designed specifically for nuclear safety applications. The TELEPERM XS system has significant nuclear operating experience. The TELEPERM XS platform has been fully qualified as an integrated platform. The TELEPERM XS system is described in TELEPERM XS Topical Report.

The overall qualification process for the TELEPERM XS system is shown in Figure 3-1. The qualification process is a two-part process: generic system qualification and specific system qualification. The qualification process for Application Software starts with the application-independent (generic) qualification process described in Section 2.1 of the TELEPERM XS Topical Report. The application-dependent (specific project) phase takes credit for all application-independent (generic) qualification activities, as noted on page 2-4 of the TELEPERM XS Topical Report.

The generic qualification process included an integration and system test phase, which is described in detail in Section 3.2.2 of the TELEPERM XS Topical Report. The system test documentation is listed in Section 8.1.1 of the TELEPERM XS Topical Report. There are no application specific requirements that affect the generic approach of integrating TELEPERM XS hardware and software. As such, the project-specific Application Software integration efforts are built on this foundation.

Application Software is developed using the TELEPERM XS SPACE tool. This tool is used to develop Function Diagrams and Network Diagrams. Function Diagrams specify the signal processing requirements for the system, which includes the signal processing with I/O devises as well as between TELEPERM XS safety processors. Network Diagrams define the hardware components of the system and their logical interconnections. Application Software code for all TELEPERM XS safety processors is

Draft

automatically generated from the Function Diagrams and Network Diagrams by the SPACE tool. Logical 'software integration' occurs at this stage. The project-specific TELEPERM XS system is developed from qualified hardware and software modules using the qualified development tools.

Physical integration of the Application Software integration with the project-specific TELEPERM XS hardware occurs during the pre-FAT stage, when the project-specific Application Software is loaded on the TELEPERM XS processors. The Application Software Installation Plan is described in Section 7.0. The project-specific FAT Plan covers the approach and activities associated with the Software and Hardware Integration.

A project-specific Software Generation and Download Procedure is issued for each project to control and document the generation of each Application Software release. It is used to control and document the download of each approved software release to the target system. This project-specific Software Generation and Download Procedure is implemented under a work order (task-letter) for each Application Software Release. The Software Generation and Download Procedure is a configuration item that is governed by the Software Configuration Management Plan. The Software Design Group implements the Software Generation and Download Procedure.

Application Software Integration Testing is performed in accordance with the Software Verification and Validation Plan, as described in Section 11.0. IEEE Std 1012-1998. The purpose of the testing is to establish that the project-specific system has been designed and implemented correctly.

Draft

7.0 SOFTWARE INSTALLATION PLAN

The Software Installation Plan describes the installation process for TELEPERM XS projects.

TELEPERM XS Application Software Downloads are controlled and documented in accordance with the project-specific TELEPERM XS Application Software Generation and Download Procedure. The purpose of this procedure is to provide instructions for the following TELEPERM XS Software processes:

- Generation and storage of the TELEPERM XS Application Software,
- Loading of the TELEPERM XS Application Software onto the Service Unit,
- Loading of the Application Software onto the TELEPERM XS Gateway,
- Loading of the System Segment Software onto the TELEPERM XS processing modules,
- Loading of the Application Software onto the TELEPERM XS processing modules,
- Loading of the L2-CP Firmware onto the TELEPERM XS communication processors,
- Loading of the H1-CP Firmware onto the TELEPERM XS communication processor,
- Preparation of the Configuration File for the TELEPERM XS communication processors, and
- Parameterization of the L2-CP Firmware for the TELEPERM XS communication processors.

The Software Generation and Download Procedure addresses the following topics:

- Ensure that the correct tools are used for the code generation and that the Application Software is correctly stored in the Software Library,

Draft

- Instructions for loading the Application Software onto the TELEPERM XS Service Unit,
- Instructions for loading the Application Software, System Segment Software onto the processing modules, preparation of the Configuration File, and loading of the Firmware components onto the communication processors, and
- Instructions for loading the applicable files from the Software Release onto the TELEPERM XS Gateway.

Software may be installed at several points during the course of the project. The first time an Application Software Release is installed is prior to FAT. After the initial installation, approved Application Software Releases containing software changes may be downloaded to the target system using the same controlled process. Each installation of TELEPERM XS Application Software is done using the Software Generation and Download Procedure, which is controlled by the TELEPERM XS Software Configuration Management Plan. The Software Design Group implements the Software Generation and Download Procedure.

The Application Software download can be performed as central download or local download. The central download utilizes the TELEPERM XS Service Unit. Local download means that the software is downloaded by directly connecting to the respective TELEPERM XS processor itself. The sequence of software download is not important for installations prior to plant operation. The Application Software installation sequence used during system operation requires special attention (e.g., the operational requirements for the system, the operational integrity of each redundancy, and the compatibility of signals exchanged between redundancies).

The TELEPERM XS SPACE Tool is used to generate the CRC checksums separately in advance of the download. The TELEPERM XS Application Software MIC files are downloaded into memory areas (segments) of the FEPROM of the respective TELEPERM XS processor. For each segment of the FEPROM, the TELEPERM XS Application Software Download method creates a CRC checksum, which is stored in the

Draft

EEPROM of the TELEPERM XS processor. After downloading, the two CRC checksums are compared to verify that the correct version of the TELEPERM XS Application Software has been loaded in accordance with the project-specific Software Generation and Download Procedure. Thus the substitution of corrupted or altered software would be detected and corrected.

The TELEPERM XS Application Software CRC checksums are periodically calculated and checked against the CRC checksums stored in the EEPROM by the TELEPERM XS Self-Monitoring Software during TELEPERM XS processor startup as well as during cyclic operation. Thus, identity and integrity of the downloaded software is verified by the TELEPERM XS System automatically after loading.

The TXS Application Software CRC checksums stored in the Flash-EPROM of the respective TXS Safety Processor can be extracted via manual request from the TXS Service Unit and compared against the CRC checksums in TXS Application Software files stored on TXS Service Unit.

Draft

8.0 SOFTWARE MAINTENANCE AND OPERATIONS PLAN

The Software Operations and Maintenance Plan which describes post-customer delivery software practices.

The Software Maintenance and Operations Plan conforms to the life cycle planning for operations and maintenance guidance of IEEE Std 1074-1995, as endorsed by Regulatory Guide 1.172. It is also consistent with the software operations and maintenance planning guidance of BTP 7-14. After the system has been turned over to the customer, software modifications and maintenance by AREVA NP are performed following this plan.

8.1 Purpose

The plan describes the activities and resources that enable AREVA NP to support and maintain TELEPERM XS safety system software after the software has been installed in a nuclear plant, such as responding to changes in customer requirements, offering upgrades, and reporting and correcting any defects or anomalies discovered in the software.

The Software Maintenance and Operations Plan identifies the process to implement a change in the software after the system has been turned over to the customer, based on one or all of the following three conditions.

- The identification and correction of software errors, performance failures, and/or implementation problems (corrective maintenance).
- Modifications to permit the software system to run in a different operating environment, with different types of data, or to incorporate new requirements (adaptive maintenance).
- Modifications to enhance performance, improve cost effectiveness, or otherwise improve the software system (perfective maintenance).

Draft

The implementation of the corrective action, including the design, implementation, testing, installation, and documentation of the corrective action, are governed by the Software Configuration Management Plan. The verification activities for the corrective action are governed by the Software Verification and Validation Plan. To ensure that the most current version of the software is utilized during the corrective action process, the Customer is consulted to ensure that all changes, including parameter changes, between the final delivered system and the current, operating system are taken into consideration. These changes are transmitted in an official document that can be utilized as a design input.

The implementation of adaptive and perfective maintenance changes shall be set up as a new TELEPERM XS project following the requirements of the AREVA NP Operating Instruction for TELEPERM XS projects. This type of maintenance requires an approved change order or purchase order and an approved set of design inputs prior to the start of work. The software development process described in the Software Program Manual would be used for these projects.

8.2 Problem Identification

When problems identified during the operating phase of the system are reported, AREVA NP assesses the potential impact on other TELEPERM XS customers and then notifies them appropriately. Problems reported to AREVA NP are evaluated to determine if the problem is a nonconformance with requirements or a condition that warrants a design change.

Any reported problems or issues are entered into the Corrective Action Program, and identify the time and date of discovery and the identity of the person making the report. Anomalies may include test deviations, system malfunctions, or inconsistencies between the software and documentation. Condition reports may also be used to initiate a request for an improvement in software performance or documentation clarity when no specific malfunction exists.

Draft

Because any error in safety system software presents the potential for common-mode failure of redundant functions, the maintenance plan requires the timely evaluation of the effects of reported problems to support equipment operability determinations as required by plant technical specifications.

Condition reports are evaluated to determine whether a nonconformance exists. A software problem or issue that is discovered during the operations and maintenance phase and proves to be a nonconformance is handled in the Corrective Action Program. If the AREVA NP evaluation determines that a nonconformance exists, AREVA NP will notify the customer that a condition report or an equivalent should be initiated by the customer's Corrective Action Program. The customer's program makes any required operability determinations.

For a customer identified problem, a nonconformance following the AREVA NP administrative procedure will be initiated. Identifying the configuration and system input and operating conditions is the first activity in responding to the nonconformance. A 10 CFR Part 21 (Reference 1) evaluation will be performed in accordance with the AREVA NP procedure for evaluation and reporting of safety significant issues (Reference 43). The nonconformance cross-references the customer's problem report number, if it is known.

If a software change is necessary to resolve a condition report or a nonconformance, a software change is made in accordance with the software configuration maintenance plan.

Software change condition reports are to be evaluated and approved following the procedures outlined in the AREVA NP Corrective Action Program. Once approved, an authorized software change is implemented following the Software Configuration Management Plan. Approved software change requests are grouped as appropriate and implemented together as a single software version release.

Draft

8.3 Analysis

The evaluation uses the Open Item report or nonconformance information and the software change authorization request validated in the modification identification and classification phase, along with system and project documentation, to study the feasibility and scope of the modification and to devise a preliminary plan for design, implementation, test, and delivery.

The analysis determines if the reported Open Item represents a nonconformance to the original design or a design change. Depending on severity, nonconformances require a root cause or apparent cause analysis to determine where the design process failed. The design process must then be repeated from that point forward, including the appropriate rework and documentation revisions to correct the problem. Design changes undergo the same process as the original design from specification to validation testing, including the required verification activities. For either nonconformances or design changes, regression analysis can be used to determine the scope of retesting.

8.4 Processing Simple Changes without Nonconformances

If the customer's requested change is dictated by a change at their plant rather than an anomaly or a nonconformance, the above process is followed with the exclusion of generating the documentation necessary to evaluate the anomaly or nonconformance. Specifically, the Open Item report is used to process the change and deliver it to the customer.

Draft

9.0 SOFTWARE TRAINING PLAN

The Software Training Plan describes a process that can be used to ensure that training needs of appropriate plant staff, including operators and I&C engineers and technicians, are met.

AREVA NP maintains the customer Software Training Plan, which includes the different facets of training, including software. The customer Software Training Plan is included as a part of the project plan. The training provided to an individual customer is based on the customer's needs as stated in the specification and implemented per the AREVA NP customer Software Training Plan. The customer's Software Training Plan and specification are referenced to produce a project Software Training Plan that meets the customer's needs and conforms to applicable regulatory standards.

The AREVA NP operating instructions that implement the customer Software Training Plan control the specifics of the training provided to the customer. These operating instructions provide the details necessary to ensure that training is developed and presented in accordance with accepted training standards. The project Software Training Plan includes the prerequisites required for each type of training listed. The project Software Training Plan provides a training schedule and specifies the instructional support required for the project. The training schedule is also loaded into the project master schedule.

Training in the use of the TELEPERM XS Application Software development tools uses versions of the software that provide the same or similar functions to be employed in the project. The training includes an introduction to the application architecture design and uses project-specific nomenclature and naming conventions.

9.1 Purpose

The Software Training Plan ensures that designated customer personnel thoroughly understand the construction, components, operation, and maintenance of the TELEPERM XS System. Training is implemented in the following manner.

Draft

- Adequate training staff is available to meet the customer training needs
- Training staff are technically and instructionally competent
- Project plans include a Software Training Plan that addresses the needs of the plant staff which:
  - includes specifics of the requirements of plant staff positions so that training can be tailored to the plant organizational structure and positional responsibilities
  - includes specifics on the timing of the various training offerings to ensure that the training is effective
  - ensures that prerequisite knowledge and skills are known to plant staff to ensure that the plant staff is technically prepared to receive instruction on the new system
  - provides a description of the systematic approach to training used by AREVA NP so that the plant staff can best incorporate the training into their accredited programs
  - provides a means for the plant staff to provide input on specific plant procedures and communication techniques used by the customer, such as the use of 3-way communications and/or phonetic alphabet, so the AREVA NP training organization can incorporate customer-specific plant cultures into the training
  - provides a means to discuss plant-specific industry experience that needs to be incorporated into the new system training
- Hardware and software that is functionally similar to the customer's system is provided for use during training
- A combination of lecture and hands-on exercises comprise the training

Draft

- Training is based on a task analysis of the system from an engineering and technical viewpoint to ensure that the objectives for specific customer groups are fulfilled
- Training materials are linked to the objectives
- Examinations ensure trainees master the objectives
- Check points are built into the process to ensure that customers have input into the training provided
- Appropriate training facilities are used for the training. AREVA NP training facilities are conducive to learning, reduce unnecessary interruptions, and provide a means of gaining hands-on experience. AREVA NP training facilities include hardware and software training aids that are similar to the customer's equipment to facilitate the learning process. Alternatively, customer training facilities may be used.

9.2 Organization

The AREVA NP customer Software Training Plan describes the training organization, which consists of dedicated training personnel and subject matter experts (SMEs). Personnel that provide training are competent in their area and have instructional training. The training organization is functionally separate from the project organization to ensure that training has the necessary independence to generate training based on needs. To ensure that training is generated specifically for the customer's system, the training organization is linked to both the project organization and the engineering organizations. The training organization has internal training processes which ensure that instructors and SMEs are technically and instructionally competent. The training department interfaces with the customer training organization through the project management organization to ensure that the needs of the customer organization are met. In addition, the training organization interfaces directly with the customer's maintenance, engineering, and operations departments as necessary.

Draft

9.3 Responsibilities

The AREVA NP customer Software Training Plan describes the training organization, which is directed by the training supervisor. The training supervisor ensures that the internal implementing procedures comply with the Software Program Manual and that the implementing procedures are maintained current with regulatory requirements. The training supervisor provides training and directs SMEs in the presentation of training when necessary. Additionally, the training supervisor provides the main interface between training, project management, engineering, and the customer to ensure that the necessary transfer of knowledge and skills is provided.

The training department is functionally separate from the project. This level of separation ensures that the required focus on training is not subsumed into the engineering or testing processes of a project.

9.4 Measurements

The implementing documents of the customer training plan ensure that feedback is gathered and incorporated into the training process. At the end of each training course, feedback is gathered by means of Level I (Kirkpatrick) instruments. In addition, the use of a customer training plan checklist encourages management oversight by the customer, ensuring the customer is aware that AREVA NP expects management observations of training. Management observations of training are incorporated with the Level I feedback from students to improve the training program.

The training processes are reviewed annually to ensure they address recent operating experience and technology advances.

Trainee evaluations are based on questions derived from the course objectives. Questions are written in a format specified by procedure, which sets the bounds for the type and format of questions. Additionally, the procedure requires that questions are checked to ensure that they are independent of other questions, do not provide clues for

Draft

subsequent questions, do not promote guessing, or otherwise compromise the validity of the examination.

The examination results and the successful completion of hands-on exercises are compiled into evaluation record sheets, which provide the data to the customer.

9.5 Procedures

The implementing documents are maintained and tracked throughout the training process. The training procedures follow the training system development process used by plant training departments. The five elements of the training process are laid out in separate procedures that guide course developers and instructors through the process:

- Analysis,
- Design,
- Development,
- Implementation, and
- Evaluation.

Each procedure has a section titled “Records” that defines the required documentation for each procedure that must be maintained.

9.6 Methods

The AREVA NP customer Software Training Plan ensures that training is carried out using hardware and software that is similar in function to the customer system. Hardware is wired to simulate the I/O and communications of an actual system. Simulated inputs can be supplied through either software or by means of a hardware cart that can be programmed to input and output analog and digital data. Techniques for power-up, power-down, and module removal and replacement can be demonstrated.

Software that allows trainees to learn the software development process is supplied. The software training uses the same development environment as that used to develop

Draft

the customer software. Customers learn the process of generating software through hands-on activities that enable them to develop programs that input and output data and generate communications between different system processors. In this manner, the entire software process from the generation of software diagrams, through the compiling, linking, locating, and finally the downloading process is presented.

9.7 Training Manuals and Materials

Operator training manuals present system responses to simulated plant conditions and manual inputs. The presentation of output indications links outputs indications to plant conditions. Operator inputs are covered in the material to show how the system reacts to manual inputs under normal and abnormal conditions. System failure modes are presented to show the outputs and indications provided for system failures. In addition to operator panel inputs, training that shows how the system responds to the inputs available at the front of the cabinets is provided. System tests that can be initiated from the graphical interface are presented, and software parameter changes that are available to operators via the graphical interface are covered.

The appropriate hardware or software engineers review the training materials to ensure that topics are presented thoroughly and accurately. Checks made against the SDD ensure an agreement between training and engineering documents. The use of standard terms identified in software user manuals ensures consistency.

Training materials provide definitions and clear, concise wording which ensure that concepts are understandable. Training materials are presented using slides and accompanying text that explains the concepts. To the extent possible, slides use graphics to explain concepts. Slide shows are reviewed to ensure that wording, descriptions, definitions, and terms are clear and consistent across the different user and training documents.

Training materials are checked against the design documents and procedures that implement this training plan to ensure that operator requirements and actions are

Draft

explained consistently. Each operator and system requirement and the automatic system responses to plant conditions are reviewed.

Draft

10.0 SOFTWARE SAFETY PLAN

The Software Safety Plan identifies the process to reasonably eliminate hazards that could jeopardize the health and safety of the public from safety-critical software.

Software safety analysis activities follow the requirements of the Software Safety Plan as part of the basic design, detailed design, testing, and installation and commissioning phases.

10.1 Purpose

To ensure that safety system software development is consistent with the defined system safety analyses, planned and documented software safety analysis activities are conducted during the basic design and detailed design phases of the software development life cycle. The analyses must ensure that:

- System safety requirements as specified in the SRS have been met correctly
- No new hazards have been introduced
- Software elements that can affect safety are identified
- There is evidence that other software elements do not affect safety
- Safety problems and resolutions identified in these analyses are documented

10.2 Management

The Technical Manager is responsible for the execution of the Software Safety Plan. Various parts of the project organization perform the analyses listed in Section 10.3. The Technical Manager ensures that these analyses are completed in accordance with the plan, with the exception of the verification and validation activities. The Technical Manager must be cognizant of the verification and validation activities.

The Project Manager has the following responsibilities for the Software Safety Plan:

- Coordinate software safety tasks within the overall context of the system safety program.

Draft

- Coordinate safety task planning with other organizational components or functions, such as development, system safety, software QA, software reliability, software configuration management, and verification and validation.
- Obtain, allocate, and monitor resources for effective implementation of the Software Safety Plan.
- Participate in audits of Software Safety Plan implementation.
- Coordinate technical issues related to software safety with the AREVA NP Inc. project Lead Software Engineer.
- Ensure training in methods, tools, and techniques used in software safety tasks for the project and Verification and Validation personnel. Ensure that the training is documented in accordance with AREVA NP administrative procedures.
- Communicate any safety concerns in accordance with the AREVA NP Corrective Action Program.

The Software Supervisor has the following responsibilities for the Software Safety Plan:

- Maintenance of the Software Safety Plan
- Overall conduct of software safety activities.
- Technical direction to members of the Software Design group for software safety activities.
- Specifying project specific training related to software safety activities in accordance with AREVA NP administrative procedures.
- Communicating any safety concerns in accordance with the AREVA NP Corrective Action Program.

Advances in software technology and processes have created new software tools that simplify the application of the software safety methodology described in IEEE Std 1228-1994 (Reference 31). The context of IEEE Std 1228-1994 is that there is a separate (or independent) group from the Software Design group that is doing coding work from a

Draft

set of functional requirements or diagrams. The NRC has not endorsed IEEE Std 1228-1994.

For TELEPERM XS Application Software, the code is automatically generated by the SPACE tool. As such, the Software Design group does not create code; instead, it is involved in many of the software safety analyses. The responsibility to produce a safe TELEPERM XS application is not separate from the responsibility to produce a quality product, or a functional product. The Technical Manager has overall responsibility for the Software Safety Plan. The Project Manager coordinates the implementation of software safety tasks for the project. Various groups perform the software safety analyses. The organization approach for the Software Safety Plan used by AREVA NP meets the intent of IEEE Std 1228-1994.

10.3 Software Safety Analyses

The AREVA NP Inc. approach to software safety analysis is based on important foundational elements.

The TELEPERM XS system is a fully integrated suite of hardware and software designed specifically for nuclear safety applications. The TELEPERM XS system has significant nuclear operating experience. The TELEPERM XS platform has been fully qualified as an integrated platform. The generic qualification process, which included independent validation testing, removes software hazards associated the generic TELEPERM XS platform. The TELEPERM XS system uses a comprehensive set of self-monitoring tests to monitor system performance for internal faults, as described in the TELEPERM XS Topical Report. The TELEPERM XS operating system has a substantial nuclear operating experience base to validate performance and provide opportunities to identify latent errors.

The software hazards associated with first-of-a-kind engineering work are minimized through the use of qualified software development tools and structured engineering analyses. The use of a qualified Function Block library provides a large experience

Draft

base for the standard modules. The use of the object-oriented automated code generation tool (SPACE) eliminates an important human error source by eliminating conventional software development and code generation. The SPACE tool eliminates both errors of translation and the introduction of complexity by engineers trying to optimize application coding. The use of the SPACE tool supports the development of high quality software with a less complex process, which eliminates many software hazards associated with manual coding.

New software engineering processes have been developed to reduce or simplify the complexity of project-specific engineering since IEEE Std 1228-1994 was issued. The software safety methodology for TELEPERM XS projects is based on the use of the pre-qualified TELEPERM XS platform and software engineering tools.

The generic TELEPERM XS foundation is supplemented with project-specific analyses that address the first-of-a-kind engineering for a TELEPERM XS project. The following sections describe the project-specific safety analyses that are performed to ensure that the Application Software satisfies the design basis safety requirements. These activities ensure the high reliability necessary for safety-related software in a safety-related system. When combined together these activities also satisfy the requirements for a software hazards analysis and meet the intent of IEEE Std 1228-1994.

10.3.1 Preliminary Hazard Analysis

A preliminary hazard analysis is intended to address preparatory activities associated with high-level system design, and the interfaces between the software and the rest of the system to identify hazardous system states or actions that can cause the system to enter a hazardous state. The generic safety assessment of the TELEPERM XS platform described in the TELEPERM XS Topical Report and the project-specific development process described in the Software Program Manual coupled with the performance of a project-specific diversity and defense-in-depth analysis satisfy the requirement for a preliminary hazards analysis.

Draft

10.3.2 Diversity and Defense-in-Depth Analysis

The diversity and defense-in-depth analysis is performed during the Basic Design Phase. The diversity and defense-in-depth analysis is performed to assess the adequacy of diversity afforded by the system design, to ensure that adequate defense-in-depth has been provided in the design, and to verify that the displays and manual controls for critical safety functions initiated by operator action are diverse from computer systems used in the automatic portion of the reactor protection and engineered safety features actuation systems. The diversity and defense-in-depth addresses residual software hazards by addressing mitigation of assumed software common mode failures.

The diversity and defense-in-depth analysis is performed in accordance with the guidance of BTP 7-19 (Reference 12). The customer usually performs this analysis, which is required by the NRC for reactor protection system and engineered safeguards features actuation system replacements. AREVA NP may or may not be involved in this analysis. AREVA NP performs this analysis for new plant designs.

The results of this analysis are incorporated into succeeding phases of software development. The methodology established in this Software Program Manual is designed to minimize the potential for software common mode failures.

10.3.3 Application Software Requirements Traceability Analysis

The Application Software requirements traceability analysis is used to document and trace software requirements through all phases of the software development. The Application Software Requirements Traceability Matrix provides supporting information for the independent verification and validation activities. The Application Software Requirements Traceability Matrix provides documented evidence that can be used by auditors to confirm that requirements and licensing commitments are met.

The traceability is a tool used by the Design organization, to ensure the completeness and the consistency of the design activities through the successive steps. It is also used by the Verification and Validation group as a tool to verify that all the requirements

Draft

have been taken into account, and to help in the test cases design for the validation activities.

10.3.4 Failure Modes and Effects Analysis

The Failure Modes and Effects Analysis (FMEA) is performed during the Detailed Design Phase. The FMEA examines the effects of random single failures on the ability of the safety system to perform its required safety functions. The FMEA conforms to the guidance of IEEE Std 379-2000 (Reference 15), as endorsed by Regulatory Guide 1.53 (Reference 3). AREVA NP meets the requirements (*shall* statements) of IEEE Std 379-2000 to establish conformance with the requirements of IEEE Std 603-1991, specifically the single-failure criterion as stated in clause 5.1. The guidance in Section 4.1 of IEEE Std 352-1987 (Reference 16) will be used for the FMEA analyses for TELEPERM XS projects, unless customer requirements specify a different format.

AREVA NP will meet the requirements IEEE Std 603-1991 clause 5.15, reliability, for those systems that have either quantitative or qualitative reliability goals established by the customer. Appropriate analysis of the design shall be performed in those cases in order to confirm that such goals have been achieved using IEEE Std 352-1987 and IEEE Std 577-1976 (Reference 17) as general guidance for the reliability analyses for TELEPERM XS projects, unless customer requirements specify a different format. The credible failure modes for TELEPERM XS hardware that are identified through the FMEA process are an input to the evaluation of reliability.

On a project-specific basis, consideration is given to performing a limited analysis of multiple random hardware and software failures, that is an “extended Failure Modes and Effects Analysis” as recommended by IEEE Std 379-2000; however, the FMEA does not need to consider the effects of a software common mode failure; instead this kind of failure is addressed by the diversity and defense-in-depth analysis discussed in Section 10.3.2. IEEE Std 379-2000 suggests in Section 5.5 that:

> Additionally, provisions should be made to address common-cause failures. Examples of techniques are detailed defense-in-depth studies, failure mode and

effects analysis, and analyses of abnormal conditions or events. Design techniques, such as diversity and defense-in-depth, can be used to address common-cause failures.

The guidance in Section 4.5 of IEEE Std 352-1987 will be used for any extended FMEA analyses performed for TELEPERM XS projects, unless customer requirements specify a different format. Extended FMEA analyses will not consider the effects of a software common mode failure because this kind of failure is specifically addressed by the diversity and defense-in-depth analysis. The consideration of multiple hardware failures consists of including failure modes or multiple failures of power supplies or other system elements that are regarded as not-credible. Such considerations shall be documented in the FMEA analysis. Extended FMEA analyses are not requirements of IEEE Std 379-2000 and are not required to establish conformance with the requirements of IEEE Std 603-1991. Instead, extended FMEA analyses, when performed, are used to provide additional insights regarding risk, reliability, or other performance objectives specified by the customer.

The FMEA includes the following characteristics:

- The Design group produces a separate document
- The FMEA determines the effects leading to failure to function as well as spurious actuation
- The FMEA is conducted at the replaceable module and component level

The FMEA ensures that the single failure requirements associated with system safety analysis requirements and assumptions are satisfied.

10.3.5 Response Time Analysis

The response time analysis is performed during the Detailed Design Phase. The response time analysis calculates the overall response time of the system, which must be lower than either the shortest response time described in the SDRD or the response time assumed in the design basis accident analyses described in the Design Control

Draft

Document for new plants or the Updated Final Safety Analysis Report for system replacements. Response time calculations consider both the TELEPERM XS hardware characteristics and the signal processing time for the software given the specified cycle time of the software.

The response time analysis ensures that the system safety analysis requirements and assumptions are satisfied.

10.3.6 Verification and Validation Activities

The verification and validation activities take place throughout the Software Life Cycle phases to provide a method of independently verifying the design that is remote and distinct from the design efforts. The verification and validation activities provide an independent process to ensure the verification of an accurate translation during each software development phase and the validation that the software product fulfills the requirements for the specific intended uses for which it was developed. Application Software requirements traceability analysis is part of the verification and validation activities.

As the project software development progresses, the Verification and Validation reports on various documents are provided in accordance with the Software Verification and Validation Plan. These reports are issued to document each of the verification and validation activities.

10.3.7 Application Software Validation Testing

Validation testing is used to validate that the right software modules have been properly used and that the functionality of the Application Software meets the software requirements and customer specifications. The Application Software validation test report summarizes the findings of the validation testing. The test report documents the compliance of the software with the prepared requirements documents and the non-compliances discovered and corrected. Validation testing simulates various TELEPERM XS malfunctions to verify that the response to these faults is as intended.

Draft

The Software Verification and Validation Plan is further described in Section 11.0. The Software Test Plan is further described on Section 13.0.

10.3.8 Criticality Analysis

The criticality analysis is performed during the Basic Design Phase and defines the SIL for all project specific software items. Review of the criticality analysis is performed during subsequent phases. The criticality analysis determines the functionality of each software module and assigns a SIL level classification for each software module based on the safety function of that module.

Non-safety software that is part of the project (e.g., TELEPERM XS Gateway or Graphic Service Monitor) must be assessed for impacts that can lead to system failures. The criticality analysis is a structured evaluation of the software characteristics for severity of impact of system failure, system degradation, and failure to meet software requirements or system objectives. The SIL assignment is reviewed as part of the verification and validation tasks to verify that the assigned SIL is appropriate for the application. Subsequent verification and validation activities are based on the SIL assignment. The verification and validation activities applied to non-safety software eliminate software hazards associated with the non-safety to safety interconnections.

10.3.9 System Testing

System testing (including FAT) is performed during the Testing Phase. System testing validates that the functionality of the system meets the design and customer requirements in the fully integrated system. The additional Application Software validation testing during system testing validates that the functionality of the Application Software meets software requirements for its intended use. A FAT is a subset of the system testing that demonstrates to the customer that the finished system meets the functional and safety requirements. A system test report (including the FAT report) is issued at the end of the testing phase. The Software Verification and Validation Plan is further described in Section 11.0. The Software Test Plan is further described on Section 13.0.

Draft

10.4 Documenting and Correcting Safety Hazards

Any safety hazard discovered in the analyses discussed above is documented using the AREVA NP Corrective Action Program and corrected as described in Section 3.4.

The measurement of the success of the Software Safety Plan is the passing of the FAT with all previously discovered safety hazards corrected in the design.

.

Draft

11.0 SOFTWARE VERIFICATION AND VALIDATION PLAN

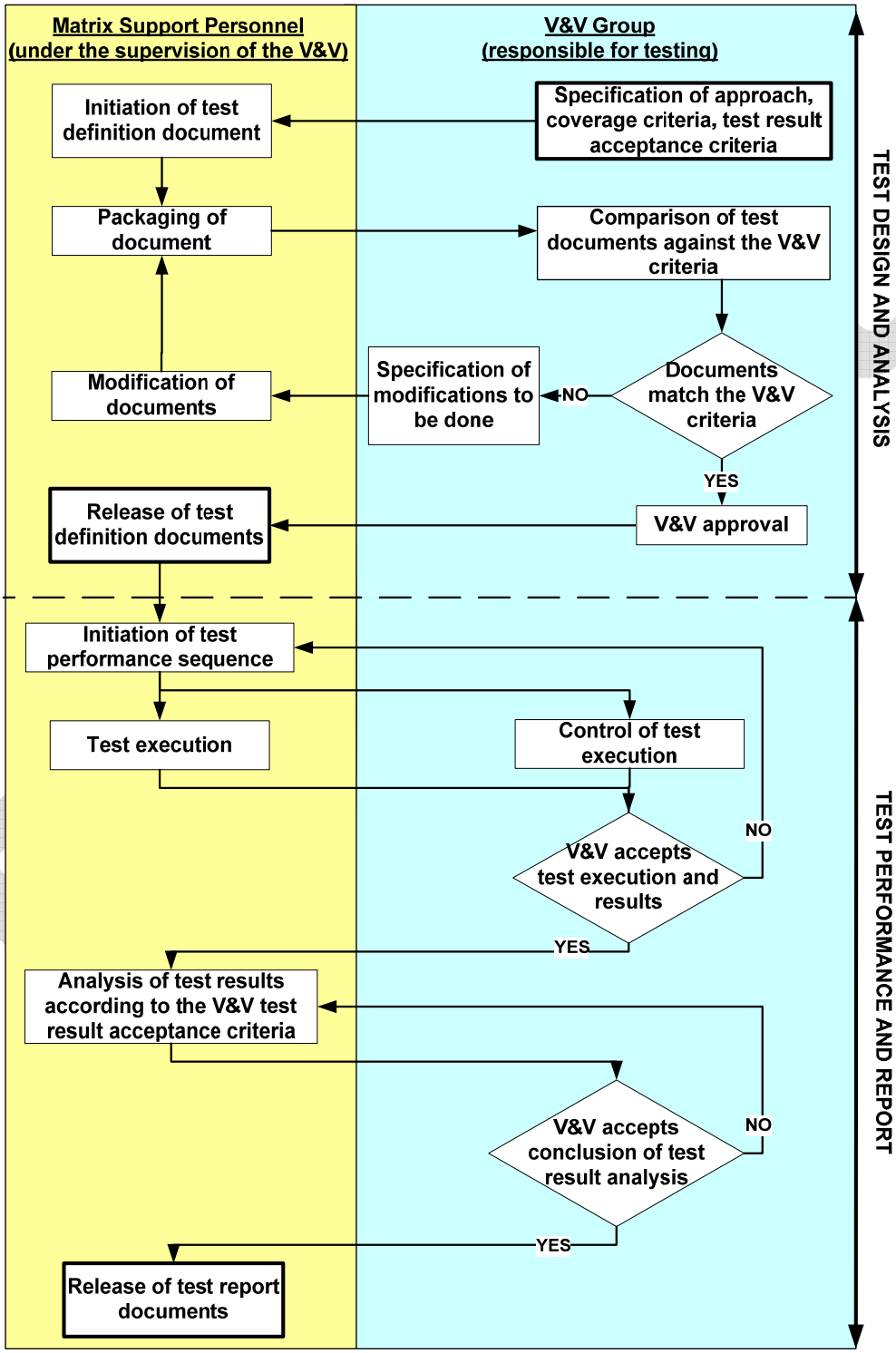
The Software Verification and Validation Plan describes the method that ensures correctness of the TELEPERM XS Application Software.

The Software Verification and Validation Plan is the systematic program of review, verification, and testing activities that are performed throughout the software development life cycle to ensure that the software documentation and products satisfy their requirements and their intended use and user needs. Personnel that are independent of those who accomplish the software design and integration perform the verification and validation activities. The Software Verification and Validation Plan describes the purpose, goals, and scope of the software verification and validation effort.

The Software Verification and Validation Plan conforms to the guidance of the applicable recommendations of IEEE Std 1012-1998, as endorsed by Regulatory Guide 1.168. One area of exception with regard to the IEEE Std 1012-1998 is the use of matrixed support personnel from the development groups to perform some of the validation test activities. The Verification and Validation group may get assistance from the Software Design group and Hardware Design group for the preparation of validation test specifications, procedures, and reports and the performance of test tasks; however, these support personnel work under the supervision of the Verification and Validation group. These support personnel shall not prepare software test documents for design work they prepared. These matrixed personnel bring special skills to supplement the validation activities. This testing method ensures that the proper hardware and software personnel are used in an integrated fashion to develop and conduct the system tests. The test document workflow and control points are shown in Figure 11-1 to ensure that the independence of validation testing is maintained.

Draft

Figure 11-1 – Test Document Work Flow and Control Points



11.1 Purpose

The Software Verification and Validation Plan specifies the activities to perform during each phase of the software development life cycle that will demonstrate acceptable levels of quality and confidence in the software being developed.

Verification and validation activities include the reviews and inspections, analyses, and tests conducted by competent individuals or groups and results in traceable documented evidence that a high level of quality and a low level of risk have been achieved.

11.2 Overview

Verification and validation processes provide an objective assessment of software products and processes throughout the software life cycle. This assessment demonstrates whether the software requirements and system requirements allocated to software, are correct, complete, accurate, consistent, and testable. Verification and validation activities also aim at detecting system and software errors, enhance management insight into process and product risk, and support the software life cycle processes to ensure compliance with program performance, schedule, and budget requirements. The Verification and Validation group performs the Activities and Tasks listed in Table 1 of IEEE Std 1012-1998 for the scope of TELEPERM XS project activities scope identified in Figures 3-1 and 3-2.

11.2.1 Organization

An organization with a specified degree of technical, managerial, and financial independence from the development organization performs the verification and validation processes.

- Technical independence requires the verification and validation effort to utilize personnel who are not involved in the development of the software. The independent verifiers are sufficiently proficient in software engineering to ensure

Draft

that software verification and validation is adequately implemented. Independent verifiers are also knowledgeable regarding nuclear safety applications.

- Managerial independence requires that the responsibility for the verification and validation effort is vested in an organization that is separate from the design organizations.
- Financial and schedule independence requires that an organization independent of the Project Manager and the software development organization control the Verification and Validation group budget. This independence prevents situations where the verification and validation effort cannot complete its activities because funds have been diverted or adverse financial or schedule pressures or influences have been exerted.

The Verification and Validation group is technically, managerially, and financially independent from the design organizations, as required by IEEE Std 1012-1998, as endorsed by Regulatory Guide 1.168. AREVA NP management is responsible for establishing financial, managerial, and technical independence for software verification and validation. This independence must ensure that the verification and validation process is not compromised by schedule and resource demands placed on the design process.

The Software Verification and Validation Plan describes the organization of the verification and validation effort including the degree of independence required for verification and validation activities.

The Software Verification and Validation Plan describes the relationship of the verification and validation processes to other processes, such as development, project management, QA, and configuration management.

The Technical Manager and the QA organization are responsible for reviewing the Software Verification and Validation Plan to assess its appropriateness to the scale and

Draft

complexity of the project. The Verification and Validation group manager is responsible for issuing and implementing the Software Verification and Validation Plan.

11.2.2 Procedures

The procedures governing the verification and validation effort have the following provisions:

- Establish the methods and procedures by which each verification and validation task is performed
- Reference the AREVA NP Open Item system for handling detected errors before products are released to the customer
- Evaluate the risks associated with each project development activity
- Provide instruction on the selection of test cases for software review requirements
- Establish the reporting requirements

11.2.3 Schedule

The Software Verification and Validation Plan defines the verification and validation activities that are planned for each phase of the software development life cycle. These activities are loaded into the project master schedule. The Software Verification and Validation Plan delineates the predecessor and successor logic for verification and validation activities by defining the required inputs for and outputs from each. The Verification and Validation engineer reviews the project schedule to ensure that its logic is consistent with the Software Verification and Validation Plan.

The Software Verification and Validation Plan summarizes the schedule of verification and validation tasks and task results as feedback to the development, organizational, and supporting processes, such as QA and configuration management. The Verification and Validation organization publishes activity reports for each verification and validation activity and summary reports, which are created at the end of each project phase. The Software Verification and Validation Plan specifies the content of

Draft

these reports. Verification and validation reports are provided to the Project Manager and the software development group on a timely basis.

The summary reports that are delivered at the conclusion of each life cycle phase include an assessment of the risk of continuing into the next phase of the life cycle by considering any Open Items or design issues described in the report. The summary report also recommends risk mitigation steps. If these risk mitigation steps are accepted by the Project Manager, these steps are added to the project schedule activities.

11.2.4 Software Integrity Levels

When differing software integrity levels are assigned within the project, the Software Verification and Validation Plan documents the SIL assignment to individual software components, such as requirements, detailed functions, software modules, subsystems, or other software tools. TELEPERM XS Application Software used in nuclear power plant safety systems is assigned SIL-4 as demonstrated by a mapping between the Software Verification and Validation Plan approach and SIL-4 as defined in IEEE Std 1012-1998.

11.2.5 Resources

The Software Verification and Validation Plan defines the tools required to perform or support verification and validation activities, the point in the development effort when each tool is needed, and the personnel training requirements to ensure the competent use of these tools, such as the software simulation test tool SIVAT. The Project Manager arranges and verifies the training and access to these tools.

11.2.6 Responsibilities

The project Software Verification and Validation Plan publishes the names and duties of the individuals assigned to the Verification and Validation group. A lead individual is assigned to each verification and validation activity. The lead individual updates the project schedule with progress information as the activity is worked. Task assignments

Draft

may be made using the project master schedule and referenced in the Software Verification and Validation Plan.

Each verification and validation activity is completed with an activity report, which is submitted to the Verification and Validation group manager for approval. At the end of each phase of the software development life cycle, the Verification and Validation engineer publishes a summary report.

The Verification and Validation group performs verification reviews of the SDRD and the traceability analysis of the SDRD into the SRS, SDD, and Software Test Plans.

The Verification and Validation group manager reviews and approves all products from verification and validation activities, including software and system test plans, specifications, procedures, and reports. The Verification and Validation group manager may delegate the approval authority to a Verification and Validation lead engineer.

The Verification and Validation group has technical competence equivalent to the Software Design group. Verification and Validation personnel are trained on the provisions of the Software Program Manual. Commensurate with their assigned responsibilities, Verification and Validation personnel shall be sufficiently proficient in software engineering to ensure that software Verification and Validation activities are adequately implemented and are knowledgeable regarding nuclear safety applications. Verification and Validation personnel shall be familiar with the design principles and features of the TELEPERM XS system. Verification and Validation personnel shall be trained in the use and the output of the SPACE tool for verification of the SPACE Function Diagrams.

Verification and Validation personnel shall be trained in the use and output of an NRC approved simulation test tool, for the preparation or verification of software validation test documents and validation testing. Verification and Validation personnel shall be familiar with acceptance test procedures, predicting test results, and the form of the generated outputs from the TELEPERM XS System for validation testing.

Draft

The Verification and Validation group is responsible for preparation of the software and system validation test documents. The Verification and Validation group may get assistance from the Software Design group and Hardware Design group for the preparation of validation test specifications, procedures, and reports and the performance of test tasks; however, these support personnel shall work under the supervision of the Verification and Validation group. These support personnel may not develop software test documents for design work they prepared.

11.2.7 Verification and Validation Methods

The TELEPERM XS Application Software verification and validation process provides a comprehensive and objective assessment of software products and processes throughout the software life cycle. This assessment demonstrates whether the software requirements and system requirements allocated to software, are correct, complete, accurate, consistent, and testable. A combination of verification and validation activities are used to detect system and software errors, enhance management insight into process and product risk, and support the software life cycle processes to ensure compliance with program performance, schedule, and budget requirements. The Verification and Validation Group performs the Activities and Tasks drawn from IEEE Std 1012-1998 for TELEPERM XS projects.

IEEE Std 1012-1998 Section 1.6 allows for customization of the task lists (e.g., combination or elimination). The following are allowed task modifications based on the TELEPERM XS technology:

- The test tasks were modified to address generic TELEPERM XS platform testing, as described in Section 13.0.
- A separate hazard analysis is not performed for TELEPERM XS technology; instead, the set of analyses performed as described in the Software Safety Plan constitute the hazard analysis for TELEPERM XS systems.

Draft

- An additional Implementation Activity criticality analysis task is not performed. Criticality is determined during the Requirements and Design activities (based on the TELEPERM XS technology attributes and the project-specific network design). No change to the criticality assignment results from the Implementation Activity.

The verification and validation tasks performed for each phase defined in IEEE Std 1012-1998 are described in the following sections.

11.2.7.1 Concept Phase Verification and Validation

The Concept Phase Verification and Validation activity represents the description of a specific implementation solution to solve the user's problem. During the Concept Phase Verification and Validation activity, the system architecture is selected, and system requirements are allocated to hardware, software, and user interface components. The Concept Phase Verification and Validation activity addresses system architectural design and system requirements analysis. The objectives are to verify the allocation of system requirements, verify the selected solution, and ensure that no false assumptions have been incorporated in the solution. The following tasks are performed during the Concept Phase Verification and Validation Activity based on the SIL classification shown by each task:

- Concept Phase Documentation Evaluation (SIL 4, 3, 2) - Verifies that the concept documentation satisfies user needs with regards to system functions and are consistent with acquisition needs, the functional requirements are feasible and testable, and overall performance can be achieved.

- Criticality Analysis (SIL 4, 3, 2) – Verifies that the SILs contained in the Criticality Analysis prepared by the development organization are properly assigned to all software items in the scope of the Verification and Validation plan.

- Hardware/Software/User Requirements Allocation Analysis (SIL 4, 3, 2) – Verifies that the Hardware Design Solutions and Software Design Solutions for

Draft

correctness, accuracy, and completeness of the concept requirement allocation to hardware requirements, software requirements, or both.

- Traceability Analysis (SIL 4, 3, 2) - Develops a preliminary Software Requirements Traceability Matrix using the requirements from the SDRD. Alternatively, the development organization may develop the Software Requirements Traceability Matrix. In such case, the Verification and Validation Group performs the traceability analysis to assure selected system architecture conceptually meets the system requirements.

- Hazard Analysis (SIL 4, 3) - Identifies potential hazards based on the conceptual design.

- Risk Analysis (SIL 4, 3) - Assesses the risk of continuing into the next phase of the life cycle by considering any Open Items or design issues and recommended risk mitigation steps.

- Security Assessment (SIL 4, 3, 2) - Assesses the safety system security capabilities required for the application and identify potential security vulnerabilities.

11.2.7.2 Requirements Phase Verification and Validation

The Requirements Phase Verification and Validation activity addresses software requirements analysis. The objectives of Verification and Validation are to ensure the correctness, completeness, accuracy, testability, and consistency of the requirements. The following tasks are performed during the Requirements Phase Verification and Validation Activity based on the Software Integrity Level (SIL) classification shown by each task.

- Software Requirements Traceability Analysis (SIL 4, 3, 2) - Performs analysis of the requirements tracing between the SDRD and the SRS to ensure correctness, consistency, accuracy, and completeness. The Software Requirements

Draft

Traceability Matrix accommodates entries to show forward and backward traceability between requirements and the resultant design and test documents.

- Software Requirements Evaluation (SIL 4, 3, 2) - Evaluates the SRS for correctness, consistency, completeness, accuracy, readability, and testability.
- Interface Analysis (SIL 4, 3, 2) - Verifies and validates that requirements for interfaces to hardware, user, operator, and other software are correct, consistent, complete, accurate, and testable.
- Criticality Analysis (SIL 4, 3, 2) - Reviews the Criticality Analysis prepared by the development organization and verifies that any required changes from the Concept Phase Verification and Validation Activity have been implemented.
- Configuration Management Assessment (SIL 4, 3) - Verifies that the reviewed documentation is controlled in accordance with the applicable Software Configuration Management Plan.
- Hazard Analysis (SIL 4, 3) - Identifies potential hazards that could result from errors and Open Items from the Software Requirements Evaluation.
- Risk Analysis (SIL 4, 3) - Assesses the risk level of hazards associated with errors and Open Items from the Software Requirements Evaluation. Assesses the risk of continuing into the next phase of the life cycle by considering any Open Items or design issues and recommended risk mitigation steps.
- Security Assessment (SIL 4, 3, 2) - Verifies that the security functional performance requirements and system configuration have been identified and any security-related Open Items from the Concept Phase Verification and Validation Activity have been addressed. Verifies that interfaces external to the system will be prevented from impacting the security of the system. Verifies that security measures for software installation and system operation have been identified, and that methods for preventing unintended code in the software have

Draft

been determined. Verifies that these considerations are contained in the system requirements.

11.2.7.3 Design Phase Verification and Validation

The Design Phase Verification and Validation demonstrates that the software design correctly, accurately, and completely reflects the software requirements and no unintended features are introduced. The following tasks are performed during the Design Phase Verification and Validation Activity based on the SIL classification shown by each task.

- Traceability Analysis (SIL 4, 3, 2) - Performs analysis of the requirements tracing between the design elements in the SDD and the requirements in the SRS.
- Software Design Evaluation (SIL 4, 3, 2) - Evaluates the SDD for correctness, consistency, completeness, accuracy, readability, and testability.
- Interface Analysis (SIL 4, 3, 2) - Verifies and validates the software design for correctness, consistency, completeness, accuracy, and testability of the interfaces with hardware, user, operator, software, and other systems.
- Criticality Analysis (SIL 4, 3, 2) - Reviews the Criticality Analysis prepared by the development organization and verifies that any required changes from the Requirements Phase Verification and Validation Activity have been implemented.
- Application Software Validation Test Plan Generation (SIL 4, 3, 2) – If integration testing is performed with a NRC approved simulation test tool, the Application Software Validation Test Plan describes the software testing objectives; features to be tested, test approach, item pass/fail criteria, responsible organization(s), the test administration, and process for test anomaly handling, equipment, staffing, and training needs, required test deliverables, and test schedule.
- System Test Plan Generation (SIL 4, 3, 2) – The System Test Plan (including FAT) describes the system and acceptance test objectives; the features to be

Draft

tested, test approach, item pass/fail criteria, responsible organization(s), the test administration, and process for test anomaly handling, equipment, staffing, and training needs, required test deliverables, and test schedule.

- Application Software Validation Test Specification Generation (SIL 4, 3, 2) - If integration testing is performed with a NRC approved simulation test tool, the Application Software Validation Test Specification addresses both test design and test cases.

- Application Software Validation Test Procedure Generation (SIL 4, 3, 2) - If integration testing is performed with a NRC approved simulation test tool, the Application Software Validation Test Procedures are developed to meet the test objectives are achieved. Testing of a requirement in SIVAT may be used as justification for excluding a test of this requirement in Acceptance Testing provided that tracing of the requirement to the SIVAT Test documentation is performed.

- Hazard Analysis (SIL 4, 3) - Identifies potential hazards that could result from errors and Open Items from the Software Design Evaluation.

- Risk Analysis (SIL 4, 3) - Assesses the risk level of hazards associated with errors and Open Items from the Software Design Evaluation. Assesses the risk of continuing into the next phase of the life cycle by considering any Open Items or design issues and recommended risk mitigation steps.

- Security Assessment (SIL 4, 3, 2) - Verifies that the security requirements identified in the system requirements specification are translated into specific configuration items in the system design, including provisions for precluding unintended additions to the code.. The security requirements shall address control of access to safety system functions and data communication with other systems. The use of passwords and possible additional access control methods shall be addressed.

Draft

11.2.7.4 Implementation Phase Verification and Validation

The Implementation Phase Verification and Validation Activity verifies the software design is correctly translated into code. The following tasks are performed during the Implementation Phase Verification and Validation Activity based on the SIL classification shown by each task.

- Traceability Analysis (SIL 4, 3, 2) - Performs analysis of requirements tracing between the Application Software Code (SPACE Function Diagrams) and the design elements in the SDD for correctness, consistency, and completeness.

- Source Code and Source Code Documentation Evaluation (SIL 4, 3, 2) - Verifies the following items: SPACE Function Diagrams were prepared correctly, that correct versions and revisions of SPACE Function Diagrams were used to assemble the Application Software release, the Software Generation and Download Procedure was followed, qualified versions of code generators have been used, and only qualified TELEPERM XS System Software components have been used.

- Interface Analysis (SIL 4, 3, 2) - Verifies and validates interfaces between SPACE Function Diagrams (or software source code as applicable for non-TELEPERM XS software products) and hardware, user, operator, software, communication, and other applicable systems for correctness, consistency, completeness, accuracy, and testability.

- Software Integration Validation Testing (SIL 4) – Validates project-specific Application Software design by performance the Software Test Plan described in Section 13.0, if integration testing is performed with a NRC approved simulation tool.

- Application Software Validation Test Report and Test Incident Report Verification (SIL 4, 3, 2) - If integration testing is performed with a NRC approved simulation test tool, the Application Software Validation Test Report documents that the

Draft

Application Software satisfies the test acceptance criteria as defined in the Test Specification.

- Hazard Analysis (SIL 4, 3) - Identifies potential hazards that could result from errors and Open Items from the Source Code and Source Code Documentation Evaluation.

- Risk Analysis (SIL 4, 3) - Assesses the risk level of hazards associated with errors and Open Items from the Source Code and Source Code Documentation Evaluation. Assesses the risk of continuing into the next phase of the life cycle by considering any Open Items or design issues and recommended risk mitigation steps.

- Security Assessment (SIL 4, 3, 2) - Assesses measures in the implementation activity are accomplished by the task for traceability analysis and the review of the simulation testing.

11.2.7.5 Test Phase Verification and Validation

The Test Phase Verification and Validation Activity consists of verifying acceptance test documentation and validating the Application Software design with testing. The objectives of this activity are to verify that the requirements in the SDRD are correctly implemented into the fully integrated system and validate project-specific system performance. The objectives of this activity is also to validate that the software requirements in the SDD are correctly implemented into the Application Software (if simulation testing is not performed). The following tasks are performed during the Test Verification and Validation Activity based on the SIL classification shown by each task.

- System Test Design Specification Generation (SIL 4, 3, 2) - The System Test Design Specification describes the software features to be tested, approach refinements, and pass/fail criteria.

- System Test Design and Test Case Specification Generation (SIL 4, 3, 2) - The System Test Design and Test Case Specification identifies the test items, input

Draft

and output specifications, environmental needs, special procedural requirements, and intercase dependencies.

- System Test Procedure Generation (SIL 4, 3, 2) – The System Test Procedures describe the test cases, test equipment, prerequisites, test setup, and detailed test procedure steps.

- Traceability Analysis (SIL 4, 3, 2) - Verifies that there is a valid relationship between the System Test Procedures and the System Test Plan, System Test Design and Test Case Specification back to the SDRD (and SDD if simulation testing is not performed for validation).

- System Test Verification (SIL 4, 3, 2) - Verifies the System Test results including the Test Item Transmittal Report, Test Log, Test Incident Report, and Test Summary Report to ensure that they demonstrate that the system satisfies the criteria of the Acceptance Test Plan and Acceptance Test Procedures. Verifies the correct versions of software were used in the System Test.

- Validation Testing (SIL 4, 3, 2) – Validates project-specific system design by performance the Software Test Plan described in Section 13.0.

- Hazard Analysis (SIL 4, 3) - Verifies that instrumentation used in acceptance testing does not introduce new hazards.

- Risk Analysis (SIL 4, 3) - Addresses the risks associated with installation of the software in the customer's facilities and recommended risk mitigation steps.

- Security Assessment (SIL 4, 3, 2) - Performs the Cyber Security testing and verify that the test results exercise and prove security features including access control and communications with external systems without reducing the reliability of safety functions.

Draft

11.2.7.6 Installation and Checkout Verification and Validation

Unless requested by the customer, no AREVA NP Verification and Validation resources are required for the listed activities. If changes to design, code, or other configuration controlled documents are made, the Verification and Validation activities shall be defined by the process described in Section 8.0. Upon the customer's request, an installation configuration audit may be performed with the customer.

11.2.7.7 Maintenance and Operation Verification and Validation

These processes are not applicable to TELEPERM XS Application Software verification and validation activities. The TELEPERM XS Application Software Verification and Validation effort is complete when the Verification and Validation Final Report for the project is issued following the Test Phase Verification and Validation Activity. Any modifications, enhancements, or additions to the software at a later date would follow the process described in Section 8.0.

11.2.8 Independent Testing and Validation

The independent validation test program performed by the Verification and Validation Group is described in Section 13.0.

11.2.9 Regression Testing

When a modification in the software design is approved after some validation testing has been accomplished, regression testing may have to be performed. Once the software has been baselined and placed under configuration control, the Verification and Validation engineer reviews proposed modifications to determine whether the proposed modifications create a need to either revise the Software Verification and Validation Plan testing program or to retest previously tested software units or modules.

11.3 Metrics

The Software Verification and Validation Plan documents the metrics to be used by the Verification and Validation group to measure the effectiveness of the software development following the guidance of IEEE Std 1012-1998 Annex E and describes

Draft

how these metrics support the verification and validation objectives. In addition, the Software Verification and Validation Plan describes the metrics used to measure the effectiveness of the verification and validation effort.

11.3.1 Metrics for Software Development Effectiveness

The Software Verification and Validation Plan defines a minimum set of quality metrics. The requirements coverage is a required metric and is defined as the fraction of requirements specified in the SRS that are traceable into the SDD and testing program. A comprehensive functional coverage is required. However, some SRS requirements may not be testable, so alternative verification means must be determined. The coverage may be recalculated by crediting approved means of alternative verification. As specified in the Software Verification and Validation Plan, the current calculated coverage is published periodically throughout the software development project.

The number of Open Items discovered by personnel, such as the design team or Verification and Validation team, in each phase of the project is the second required metric for software development effectiveness. For software development to be considered effective, this number should be trending down for the development.

11.3.2 Metrics for Verification and Validation Effectiveness

The set of metrics below for verification and validation effectiveness are the minimum set and can be expanded upon at the discretion of the Verification and Validation team management. The Software Verification and Validation Plan covers the use of the following metrics in more detail:

- Total number of verification and validation Open Items in the Open Item backlog as a function of calendar time
- Number of project Open Items discovered by verification and validation compared to the total number of Open Items
- Severity and risk statistics associated with errors and Open Items discovered during verification and validation activities

Draft

- Number of test anomalies discovered during independent verification and validation testing

11.4 Verification and Validation Reports

Verification and validation reports document the verification and validation activities. The report includes the review of documentation requirements, evaluation criteria, error reporting, and anomaly resolution.

Verification and validation reports include both positive and negative findings and summarize the actions performed as well as the methods and tools used.

Draft

12.0 SOFTWARE CONFIGURATION MANAGEMENT PLAN

The Software Configuration Management Plan describes the method that maintains the project-specific TELEPERM XS Software in a controlled configuration.

12.1 Introduction

The AREVA NP corporate configuration management policies and procedures (References 33 - 37) provide the context in which the Software Configuration Management Plan for the TELEPERM XS system operates. Software configuration management is the process for identifying software configuration items, controlling the implementation of and changes to software, recording and reporting the status of changes, and verifying the completeness and correctness of the released software.

12.1.1 Purpose

The Software Configuration Management Plan defines the activities, methods, and resources needed to implement the requirements of the Quality Management Manual with respect to design control activities for safety system software. The Software Configuration Management Plan conforms to the guidance of IEEE Std 828-1990 and IEEE Std 1042-1987 (Reference 27), as endorsed in Regulatory Guide 1.169. The overall AREVA NP approach to configuration management of the TELEPERM XS platform, TELEPERM XS projects, and the project-specific Application Software meets the intent of IEEE Std 828-1990 and IEEE Std 1042-1987.

12.1.2 Scope

Section 4.0 of this manual provides an overview of the TELEPERM XS software development process. The Software Configuration Management Plan lists the products which are configuration items, which vary only slightly from project to project. A project plan or specific Software Configuration Management Plan would cover a generic Configuration Item list variation, but the same configuration control process applies to any additional configuration items. The control of configuration items is a formal process and applies to the project phases as described in Section 12.0 of this manual.

Draft

The software configuration control begins in the first phase of the project as described in Section 12.0 and continues until the system is released to the customer. Section 2.0 defines key terms.

12.2 Management

12.2.1 Organization

The Technical Manager is responsible for the implementation of the Software Configuration Management Plan. The Software Supervisor is responsible for implementing various configuration management activities. The organization is as described in Section 3.2 above.

Certain activities described herein (with the exception of the Verification and Validation tasks) may be performed by any group; however, these software configuration management activities must be documented and reviewed by the Technical Manager, who remains responsible for the correct implementation of the project-specific Software Configuration Management Plan.

To ensure conformance to project schedule and contractual requirements, the Project Manager and the manager of the department have final approval of change requests.

The Software Librarian is designated in writing and is responsible for the activities described in Section 12.3.4.

12.2.2 Responsibilities

The Software Configuration Management Plan specifies the person or group responsible for the successful completion of each software configuration management task. It specifies the individual that has the authority to release any software, data, or documents for revision and the individual that has the authority to release any software, data, or documents for operation after revision is complete.

Upon release to the customer, software configuration items are entered into the customer's configuration control program. This process normally begins with the

Draft

creation of a software configuration package, which documents the attributes defined in the customer program.

To satisfy the needs of the Software Configuration Management Plan, the Project Manager meets with the customer's software configuration management personnel to determine the information to accompany each release of TELEPERM XS software configuration items. Any special agreed upon requirements are incorporated into the project plan.

The version numbers of released software components are tracked and published, and document revisions are transmitted to the customer. Configuration status accounting reports are published periodically at the frequency established in the project plan.

The Software Supervisor has overall responsibility for the software configuration management for TELEPERM XS projects. After the completion of the design process and any engineering tests in the simulation environment, the Software Supervisor releases the TELEPERM XS Application Software for installation and validation testing. After successful completion of the software validation tests and the system tests (including FAT), the Software Supervisor releases the final version of the TELEPERM XS Application Software to the customer for installation and commissioning in the plant.

Personnel assigned to work on TELEPERM XS software development projects are trained in the requirements of the Software Quality Assurance Plan and the Software Configuration Management Plan and skilled in the use of the tools described below as required by their individual job functions.

Archive copies of each version of the TELEPERM XS system and Application Software used or created for the project is kept in a software library that is fire-proof. Additional back-up copies may be created for disaster recovery as required per the project plan. Storage media is clearly and indelibly marked for easy and unambiguous identification.

The versions of TELEPERM XS tools (e.g., SPACE or any NRC approved simulation test tool) used on each release of the Application Software are controlled and recorded.

Draft

The Verification and Validation engineer uses the methods and tools described in the Software Configuration Management Plan to independently verify the configuration of the installed TELEPERM XS software and document this verification and validation activity in accordance with the project-specific verification and validation plan.

12.2.3 Configuration Control Boards

AREVA NP uses Configuration Control Boards for the development of TELEPERM XS Application Software for control of changes to functional requirements after the Application Software has been developed to the baseline stage. The overall AREVA NP approach to configuration management of the TELEPERM XS platform, TELEPERM XS projects, and the project-specific Application Software meets the intent of IEEE Std 828-1990 and IEEE Std 1042-1987.

Based on the discussion in IEEE Std 1042-1987, the high-level Configuration Control Board is not directly applicable to TELEPERM XS projects. Instead, the software-related decisions contemplated at this level are handled generically for the TELEPERM XS platform. The TELEPERM XS platform configuration management process utilizes a Configuration Control Board.

TELEPERM XS projects are built using the qualified TELEPERM XS platform, which is a fully integrated suite of hardware and software designed specifically for nuclear safety applications. Application software for TELEPERM XS projects is developed using the SPACE tool using qualified hardware and software modules. This tool is used to develop Function Diagrams and Network Diagrams. Function Diagrams specify the signal processing requirements for the system. Network Diagrams define the hardware components of the system and their logical interconnections. Application software code is automatically generated from the Function Diagrams and Network Diagrams by the SPACE tool. The Application Software is a direct outcome of this design process; it is not developed separately.

The intent of the high-level project-related review is addressed the by the routine project management meetings established in the project plan. These project meetings include

internal project meetings, customer interface meetings, and management oversight meetings and involve the project stakeholders.

All changes to project Application Software are tracked via the Open Item process, which requires an evaluation of affected documents and software changes. Software errors that are conditions adverse to quality are also processed in the Corrective Action Program.

A Configuration Control Board to address functional changes to the Application Software for a TELEPERM XS project is convened in addition to the project meetings, Design Review Boards, and Open Item process. Members of the Configuration Control Board include the project team members that participate in other forums and interact with each other on a daily basis but also include Project Management and Quality Assurance representatives.

The Software Configuration Management Plan specifies the organizational responsibilities, configuration management controls, change management controls, and interface controls. The Technical Manager is responsible for the implementation of the Software Configuration Management Plan.

Other policies, directives, and procedures do not place external constraints on the plan.

12.3 Software Configuration Management Implementation Activities

The Software Configuration Management Plan identifies the generic configuration items for a TELEPERM XS project of AREVA NP. It identifies the externally supplied configuration items supplied by AREVA NP GmbH and other software vendors and the configuration items generated by AREVA NP.

12.3.1 Configuration Item Naming and Labeling

The Software Configuration Management Plan specifies the activities and conventions for uniquely identifying and naming the configuration items. It specifies the procedures for placing items under configuration control. It describes the method for keeping data

Draft

files and tables synchronized with the software that uses them and for keeping software and its associated documentation synchronized. It specifies the procedure for associating source code with the derived object code and executable modules.

The convention used to identify and label configuration items must be used consistently throughout the project.

The FunBase design tool administers the naming of software modules, parameters, signals, data tables and other entities in the design so that each is uniquely and consistently named and properly connected in the Application Software.

Each configuration item is labeled unambiguously so that a basis can be established for controlling and referencing the configuration items defined in the Software Configuration Manual Plan.

12.3.2 Protection of Configuration Items

Procedures exist for protecting configuration items. The plan describes how configuration items are stored, handled, retained, and shipped. Tracking systems exist for managing configuration items so that the revision history of each configuration item may be retrieved and so that the latest revision of each configuration item may be easily identified.

AREVA NP implementing procedures will describe the provisions for backup and disaster recovery.

The Software Configuration Management Plan governs the control and retrieval of qualification information associated with the software designs and code, software confirmation audits, and status accounting. The following items will be controlled:

- Code
- Exact versions of support software used in development
- Libraries of software components essential to safety
- Code used in testing

Draft

- Databases and software configuration data

In accordance with the AREVA NP QA program, documentation is stored as a quality record in the AREVA NP Records Management System. This includes system configuration documentation.

Section 12.3.4 lists the software objects that are stored in the software library.

12.3.3 Purchased Software

Purchased software is acquired using the AREVA NP procurement process as described in the AREVA NP Quality Management Manual.

The Software Configuration Management Plan includes a description of the process used to maintain and track purchased items, such as software tools used to make the final product.

Purchased or re-use software is evaluated to determine the adequacy of the software. The level of evaluation is determined through the following classifications:

- Commercial-off-the-shelf development tools, such as the compiler or linker, loader, and commercial-off-the-shelf project management tools, such as Microsoft Office or IBM RequisitePro™, do not require extensive verification and validation or testing to qualify their use because the end product is extensively tested and the tool is not used in the online operation of the system. Compilers, linkers, and loaders are maintained under configuration control.
- TELEPERM XS System software and tools developed by AREVA NP GmbH under its software QA program and to be incorporated into the delivered product as-is without modifications are to be accompanied with a certificate of conformance.
- Interfacing software, such as TELEPERM XS Gateway software or test machine software, is subjected to verification and validation reviews commensurate with their relative importance to safety, assigned SIL, and taken into the software

Draft

configuration management baseline and maintained under the configuration controls of the Software Configuration Management Plan.

12.3.4 Software Library

The software library is administratively controlled by the Software Librarian. The controlling Operating Instruction describes the methods for storing electronic files. Safety-related software is stored in the software library in a locked, fire proof safe. Only the software librarian can access this software. The standard set of software stored in the Software Library for each TELEPERM XS project includes all revisions of the following software:

- TELEPERM XS System Software procured for the project,
- TELEPERM XS Application Software produced for the project,
- TELEPERM XS Gateway Software produced for the project, and
- TELEPERM XS Graphic Service Monitor (GSM) Software produced for the project.

The Operating Instruction describes the methods for moving configuration items in and out of the library and requires the librarian to check any incoming software for viruses before placing the software into the library.

The Operating Instruction also provides a method for shipping configuration controlled software to the customer.

12.3.5 Configuration Baseline Management

Configuration baselines are established for each project phase and define the basis for further development, allow control of configuration items, and permit traceability between configuration items. The baseline is established before the set of activities can be considered complete.

Draft

Baselining of a document or product is the term used when formal review has been performed as defined by the Software Configuration Management Plan and the document or product is approved for use in the next development activity or release to the customer. At this point, the document or product is assigned a revision or version number, and further changes are documented by reissue and incrementing the revision or version number.

Formal review and agreement means that the responsible management has reviewed and approved a baseline. Baselines are subject to change control.

The review and approval signature of the developing engineer and the project engineer are the minimum level of approval for baselining. The project plan defines any required reviews beyond this minimum level.

Once a baseline is established, it is protected from change. Change control activities are followed whenever a derivative baseline is developed from an established baseline. A revised baseline is traceable to the baseline from which it was established, the design outputs it identified, or to the activity with which it is associated.

12.3.6 Change Management and Configuration Control

Software changes are initiated from anomaly reports, problem analyses, and statistical monitoring of software performance. Software changes are initiated by entering an Open Item in the Open Items tracking system described in Section 3.4.

Design changes are subject to design control measures commensurate with those applied to the original design. This encompasses the re-examination of any appropriate safety analysis related to the change. Any proposed design change is evaluated for impact on the project safety analyses as described in the Software Safety Plan.

Change control preserves the integrity of configuration items and baselines by providing protection against their change. Any change to a configuration item causes a change to its configuration identification, which is indicated with a version number and attached change date.

Draft

Changes to baselines and to configuration items under change control are recorded, approved, and tracked. If the change is due to a problem report or a customer request, traceability exists between the problem report or request and the change. Software changes are traced to their point of origin, and the software processes affected by the change are repeated from the point of change to the point of discovery. Regression testing is used to test changes made to software. Regression testing ensures that the modifications do not produce unintended adverse effects and that the modified software still meets the original requirements.

12.3.7 Modification Procedures

Software modifications installed in the test field follow written procedures.

After the successful completion of the modification design process, a download strategy is determined as a pre-condition of the download release. The download can be performed as central download or local download. The central download utilizes the TELEPERM XS Service Unit. A local download is accomplished by directly connecting to the respective processor itself.

Downloads are documented. Each modification of the TELEPERM XS Application Software is recorded in a copy of the software generation and download procedure to confirm that the methods and procedures defined in this document are followed and maintained.

After downloading and resetting the processor, the CRC checksums are checked to ensure that the code was correctly loaded into the right processor.

If the CRC checksums do not match, the download was not correctly executed and is repeated. In addition, this event is documented as an anomaly and corrected and closed per the Open Item system.

12.3.8 Software Download Control

Software may be installed at several points during the course of the project. The first time an Application Software Release is installed is prior to FAT. After the initial

Draft

installation, approved Application Software Releases containing software changes may be downloaded to the target system using the same controlled process. Each installation of Application Software is done using the project-specific Software Generation and Download Procedure.

A project-specific Software Generation and Download Procedure is issued for each project to control and document the download of each approved software release to the target system. The Software Generation and Download Procedure is a configuration item that is governed by the Software Configuration Management Plan.

The steps in Software Generation and Download Procedure ensure that the correct version of the SPACE tool is used for the code generation and that the correct System Segment Software components are installed for the Application Software. The TELEPERM XS SPACE Environment and associated tools configuration is checked and documented with each Software Release created by the steps in this document.

12.3.9 Modifying Changeable Parameters

The TELEPERM XS Service Unit can be used to modify the changeable parameters. For configuration management purposes, these changes are considered design changes and are tracked and authorized using an Open Item.

In the course of making modifications to changeable parameters, the online database is updated to ensure consistency with archived plant documentation, animated function diagrams, and current software status. In cases where the online changeable parameter is copied over into the new specification, an Open Item for software modification is opened and processed following the change control procedures described above.

After a parameter change, the list of changeable parameters has to be regenerated. This list is compared with the online database of the TELEPERM XS Service Unit to verify that only the intended parameters were changed.

Draft

12.3.10 Problem Reporting

Open item forms describe anomalous and inconsistent software and documentation. Errors, anomalies, and other problems are reported and classified using the Open Items tracking system described in Section 3.4.

Problem reports that require corrective action involving redesign invoke the AREVA NP Corrective Action Program. Section 3.4 describes procedures for tracking problem reports and for ensuring that each reported problem is resolved correctly.

A change request, whether prompted by a problem or a customer request, is initiated as an Open Item as described in Section 3.4. The use of the Open Item system ensures that each request is evaluated for extent, risk, and reportability. The Open Item tracking system is also the primary source of statistical information for software quality metrics.

The Software Configuration Management Plan describes the information required to approve a change request and ensures the control of software design changes. The relationship of software configuration management to other change control procedures, such as verification and validation, anomaly handling, and maintenance, is described. Verification and validation anomaly reports are handled as Open Items and identified in the Open Items tracking system as described in Section 3.4.

12.3.11 Status Accounting, Reviews and Audits

Status accounting takes place for each configuration item prior to the completion of each phase of the project. The status accounting documents configuration item identifications, baselines, problem report status, change history and release status.

Software Configuration Items List status accounting reports are prepared at the end of each phase, as required by the Software Configuration Management Plan.

During the progress of the projects, three types of status accounts, reviews, or audits are performed and periodic management reviews are held: Code Configuration Document, Managerial Reviews, and QA Audits.

Draft

Code configuration documents are issued whenever system or Application Software is generated. Managerial reviews are performed periodically at the discretion of the Technical Manager. QA audits are performed annually and can be performed more often if a need is determined.

Draft

13.0 SOFTWARE TEST PLAN

Software Test Plan, which describes the purpose and scope of the TELEPERM XS Application Software testing activities.

The overall qualification process for the TELEPERM XS system is shown in Figure 2-1. The qualification process is a two-part process: generic system qualification and specific system qualification. The application-independent qualification of the TELEPERM XS System includes the type test of the hardware and software components and the plant-independent system test. This qualification testing program is described in Section 3.2 of the TELEPERM XS Topical Report. The generic TELEPERM XS platform software and hardware integration is subject to the generic qualification process described in the TELEPERM XS topical report. The generic qualification approach provides a very high degree of validation independence commensurate with the importance of generic system qualification. The generic qualification work provides the foundation for the project-specific system testing, including FAT.

The tests were carried out by independent third party organizations and successful completion is documented by means of test reports and certificates. The application-dependent (specific project) phase takes credit for all application-independent (generic) qualification activities. The characteristics of the type-tested components are regarded as system invariants; their specification and function were tested independently of the application, there is no need to repeat the tests for each TELEPERM XS project.

The project-specific tests include configuration tests, tests of the interfaces to field signals and other I&C systems, failure behavior, and validation of performance data (e.g., response time). The project-specific tests also include validation tests of the Application Software (i.e., I&C functions) required by IEEE Std 1012-1998. The project-specific tests also address project-specific characteristics and functions not covered by the plant-independent tests, such as:

Draft

- Hardware and software configurations (i.e., correct settings of plug-in jumpers, DIP switches, etc.),
- Signal status processing (i.e., correct handling of faulty signals), and
- Plant-specific implementation of access control and cyber security features.

13.1 Alignment with IEEE Std 1012-1998 Testing Activities

IEEE Std 1012-1998 describes four testing activities:

- Component Testing: Testing conducted to verify the correct implementation of the design and compliance with program requirements for one software element (e.g., unit, module) or a collection of software elements. (Clause 3.1.3)
- Integration Testing: An orderly progression of testing of incremental pieces of the software program in which software elements, hardware elements, or both are: combined and tested until the entire system has been integrated to show compliance with the program design, and capabilities and requirements of the system. (Clause 3.1.10)
- System Testing: The activities of testing an integrated hardware and software system to verify and validate whether the system meets its original objectives. (Clause 3.1.26)
- Acceptance Testing: Testing conducted in an operational environment to determine whether a system satisfies its acceptance criteria (i.e., initial requirements and current needs of its user) and to enable the customer to determine whether to accept the system. (Clause 3.1.1)

IEEE Std 1012-1998 shows a progression of test activities (i.e., component, integration, system, and acceptance testing) occurring during the development process (i.e., design, implementation, and test activities).

Draft

The combination of TELEPERM XS generic qualification testing and project-specific testing addresses all of the testing activities in IEEE Std 1012-1998, as shown in Table 13-1.

The terms “software modules” and “subsystems” are generally used to describe groups of software objects. This usage applies to a collection of function blocks in the SPACE diagrams that perform a particular function or set of functions within the Application Software. These general terms could also be interchanged with the terms “Function Diagrams” and “Function Diagram Group Modules” as used in the TELEPERM XS Topical Report.

13.2 Application Software Validation Testing with Simulation Test Tool

Application Software integration and functional testing can be performed with a NRC approved simulation test tools to satisfy IEEE Std 1012-1998 validation requirements. Validation testing in a summation environment can be one of the layers of validation testing that is used to ensure Application Software quality.

Testing with in the simulation environment with a NRC approved simulation tool can serve as module or unit testing (i.e., FD or FDG testing). It can also serve as integration testing of the TELEPERM XS Application Software (i.e., testing of the Application Software for all TELEPERM XS modules working together) within the limitations of simulation. Additional testing is performed as part of the manufacturing tests to address the limitations of simulation testing.

The benefit of Application Software validation testing with a simulation test tool is the early detection of faults. A balance is drawn between performing Application Software validation testing during FAT later in the development process (e.g., to support customer QA observation and monitoring) and performing Application Software validation testing in a simulation environment earlier in the process. IEEE Std 1008-1987 (Reference 24) recognizes that:

Draft

Table 13-1 Alignment with IEEE Std 1012-1998 Test Activities

IEEE Std 1012-1998 Testing Activity	Generic TELEPERM XS Testing	Project-Specific Testing
Component Testing	**X** (hardware and software type tests, including Function Blocks)	**Not Applicable** (based on use of qualified hardware and software modules)
Integration Testing	**X**	Application Software: Integration testing using a NRC approved simulation test tool for integration of FB modules and FDs **Optional X** (see Note 1)
		System Components: Pre-FAT prerequisites and procedure dry runs (manufacturing tests)
System Testing	**X**	**X** (integrated in systems testing, including FAT, based on use of qualified system components and development tools)
Acceptance Testing	**Not Applicable**	

Legend: X indicates alignment with IEEE Std 1012-1998 testing.

Note 1 – Application Software integration and functional test cases performed in accordance with any NRC approval for a simulation test tool can be used to satisfy IEEE Std 1012-1998 validation requirements.

> There are significant economic benefits in the early detection of faults. This implies that test set development should start as soon as practical following availability of the unit requirements documentation because of the resulting requirements verification and validation. It also implies that as much as practical should be tested at the unit level. (Paragraph B2.4)

The early detection of Application Software faults through validation testing in a simulation environment can reduce project risks earlier in the development process.

Draft

The Software Validation Test Plan is prepared if simulation testing is used for validation testing. Software Validation Test Plan ensures that the functional requirements of the software design detailed in the SDD are properly implemented in the SPACE application. The comprehensiveness of the testing effort should ensure that all functionality defined in the SDD is tested.

The Software Validation Testing is performed under the direction of the Verification and Validation Group. The Software Validation Test Plans, Specifications, Procedures, and Reports are prepared in accordance with the Software Verification and Validation Plan and 10 CFR Part 50 Appendix B QA requirements.

Simulation Test Specifications, if used for validation testing, will incorporate the Test-Design Specification and Test-Case Specification into a single document and conform to IEEE Std 829-1983 and IEEE Std 1008-1987. Test Procedures are prepared for each Test Case and contain test scripts that implement the test cases defined in the Test Specifications. The Test Procedures will conform to IEEE Std 829-1983 and IEEE Std 1008-1987.

The Verification and Validation Group uses the Software Requirements Traceability Matrix to ensure that software functional requirements (from the SDD) have been tested in the simulation tests. The Verification and Validation Group independently verifies that the Application Software versions being tested match those listed in the Software Configuration Management Plan.

The validation tests carried out in the simulation environment are designed such that they can be repeated to support validation of future changes to the TELEPERM XS Application Software.

13.3 System Validation Testing in the Test Field

The project-specific system test in the test field, including FAT, fulfills the requirement for system integration and acceptance testing. This testing includes test cases to address the limitations of any NRC approved simulation test tool used for validation

Draft

testing. Additional Application Software integration and functional test cases to validate engineering I&C functionality are added to the scope of system testing for the case where simulation testing is not used to satisfy IEEE Std 1012-1998 validation requirements for the project Application Software. The FAT is a formal project milestone that is attended by both AREVA NP QA and customer personnel.

The generic TELEPERM XS platform software and hardware integration is subject to the generic qualification process described in the TELEPERM XS topical report. The generic qualification approach provides a very high degree of validation independence commensurate with the importance of generic system qualification. The generic qualification work provides the foundation for the project-specific system testing, including FAT.

System testing (including FAT) is performed during the testing phase. System testing validates that the functionality of the system meets the design and customer requirements in the fully integrated system. Application Software validation testing can also be performed as a step of the system testing (i.e., on the target system) if it was not performed in the simulation environment. The additional Application Software validation testing during system testing validates that the engineering I&C functionality of the Application Software meets software requirements for its intended use. A FAT demonstrates to the customer that the finished system meets the functional and safety requirements. A system test report (including the FAT report) is issued at the end of the testing phase.

The System Test, including FAT, is performed under the direction of the Verification and Validation Group. The System Test Plans, Specifications, Procedures, and Reports are prepared in accordance with the Software Verification and Validation Plan and 10 CFR Part 50 Appendix B QA requirements.

The Verification and Validation Group uses the Software Requirements Traceability Matrix to ensure that system functional requirements (from the SDRD) have been tested in the system tests. Similarly, the Verification and Validation Group uses the Software

Draft

Requirements Traceability Matrix to ensure that software requirements (from the SDD) have been tested if Application Software integration and functional testing is performed during system testing. The Verification and Validation Group independently verifies that Application Software versions being tested match those listed in the Software Configuration Management documentation.

In the test field, the system is installed in the configuration planned for plant, and its interfaces and system functionality are tested. The integration test serves to prove that the complete system (hardware and software including non-TELEPERM XS systems) meets the requirements described in the SDRD. Proof of the correct interfaces and system functionality of the different I&C part systems and the system behavior in case of failures must be provided, particularly with respect to those part functions and characteristics which cannot appropriately be tested in the simulation environment (e.g., power supply, cabinet signaling devices, complex relay circuits and analogue circuits, communication with non-TELEPERM XS systems, time-related requirements, etc.).

The tests are carried out on a special ERBUS test computer which specifies signals to the I&C via an interface and records the output signals. The service device provides an additional possibility for specifying and querying signals.

The signals in the special test computer are specified by scripts and/or a plant simulator. In the second case, the test can be carried out as closed-loop test.

If necessary, real components are connected to the system in the test field (e.g. neutron flux sensor), or control room devices and displays.

13.3.1 System Validation Test Scope

The System Validation Test Scope includes the following:

- Electric function of the cabinets including the correct function of the I/O channels,
- System characteristics (e.g., dynamic behavior, system loads, and restart),
- System behavior in case of failures of the components and system parts,

Draft

- Correct implementation of the system level I&C task and verification of the interfaces, and

- Test of the specified function of the overall system under selected, simulated operation and failure conditions

The tests are divided in several phases.

- FAT Prerequisite Tests (Manufacturing Tests) - These tests provide integration testing of the TELEPERM XS System hardware components.

- Test Field Installation Verification - Test field installation configuration verification checks all connection and setup information required to prepare for the FAT. This information includes such items as the power and grounding connections for the TELEPERM XS cabinets, the network configuration of the equipment (including the test equipment), and the connections of the test machines and data acquisition equipment.

- Equipment Power-Up Test - Equipment power-up testing is performed to validate the configuration and functional design of the system power distribution. The test includes functional testing of cabinet power to verify the appropriate voltage is present and distributed to the correct terminal points throughout the equipment as designed. This test also verifies the functionality of the redundant power supplies. This test is performed on a cabinet by cabinet basis.

- Software Generation and Download - The Software Generation and Download Procedure is required to be run before proceeding to the remaining prerequisites and FAT. The purpose of this procedure is to provide instructions for loading software on the TELEPERM XS System:

- Communications Test - The bus connection with the service device is checked by requesting the status of all computers. The same process is used

Draft

for the TELEPERM XS processors. Each TELEPERM XS processor is reset using the Reset function.

- Cabinet Alarm Monitoring Test - Cabinet Alarm Monitoring testing will verify that the internal monitoring alarms of the cabinet monitoring system (including internal indications and signals) operate as designed.

- I/O to Field Test - This test verifies the correct cabinet internal wiring from the terminal points to the TELEPERM XS input modules and from the TELEPERM XS output modules to the terminal points. This internal wiring includes any hardware logic that may exist. Manual injection of test signals to the system inputs and manual recording of the system outputs at the field terminals is the method used for this test. This test utilizes calibrated test equipment and is responsible for validating the accuracy of the supplied equipment. When available, actual key switches, pushbuttons, lights, etc. should be used to simulate inputs and outputs.

- I/O to Test Machine Test - The purpose of this test is to validate the correct hardware and software configuration of the TELEPERM XS ERBUS test machine interface to TELEPERM XS system hardware Inputs and Outputs. This means all connections of the test machine to and from the TELEPERM XS system are checked for correct assignment to the corresponding (internal cabinet) terminal, correct assignment of signal IDs, and the correct simulation of the measuring ranges (analog) / signal levels (binary).

13.3.2 Test Field Validation Tests (System and Acceptance Tests)

The project-specific test field validation tests satisfy IEEE Std 1012-1998 system and acceptance testing requirements for the project-specific TELEPERM XS System. Additional Application Software integration and functional test cases to validate engineering I&C functionality are added to the scope of system validation testing for the case where simulation testing is not used for Application Software integration and functional testing to satisfy IEEE Std 1012-1998 validation requirements.

Draft

13.3.2.1 Test of the Required I&C Functionality

The test comprises the function diagram modules of the different TELEPERM XS processors, the connections to the I/O units and the wiring of the interfaces to other I&C systems (e.g. alarm system, gateways, and reactor output control).

If no or only a partial simulation testing is carried out, the I&C functions not tested and a sufficient overlap will have to be tested.

If the I&C functions are completely validated in the simulation environment, representative test cases of the test scope of the simulation tests are selected and to be carried out with the same simulation test scripts (converted in the test field syntax). The selection criteria for representative test cases are:

- Each TELEPERM XS processor has to be covered by at least one test case,
- Test cases with specific hardware dependencies (e.g. time-related correlation of measuring signals like neutron flux measurement and the appropriate measuring range),
- Selected test cases containing functions which are spread out across several TELEPERM XS processors (due to the asynchronous working method of the TELEPERM XS processors), and
- Selected test cases with complex functions (because of performance test).

13.3.2.2 Test of Fulfillment of the Process Engineering Requirements

This is a test whether the complete TELEPERM XS system (hardware and software) fulfils the I&C requirements. The aim of this section is the validation of the I&C functionality based on the process engineering tasks specified in the Software Requirements Specification.

For this purpose, representative simulation test cases (open-loop test) are selected and repeated. Evidence is provided that the safety I&C meets the process engineering

Draft

requirements for design basis accidents and for relevant operation and failure conditions. The test cases refer to the complete system, thus the interactions between the I&C functions are included in the test. All required actions must be carried out while avoiding spurious tripping. If technically possible and useful, the interactions between the system to be tested and other systems (possibly of a lower I&C category) are tested as well.

If no or only a partial simulation testing is carried out, the I&C functions not tested and a sufficient overlap will have to be tested.

Tests of control functions whose dynamic behavior is not significantly affected by the feedback of the system can be carried out as open-loop tests.

If significant dynamic feedbacks with the plant have to be considered, which were not tested in the simulation environment, then closed-loop tests should be used, however, this is subject to the availability of dynamic plant models. Otherwise simplified models can be used.

13.3.2.3 Hardware Failure Tests

The test program is defined together with the customer. The points to be tested within the scope of the individual error tolerance include:

- The failure of a TELEPERM XS processor must not affect the execution of the I&C function,
- The failure of I/O modules must not affect the execution of the I&C functions (can be carried out in terms of software via the service unit) or it can be covered as worst case by computer failure. New modules with parameterized output settings are taken into account,
- The failure of a bus connection must not affect the execution of the I&C function,

Draft

- The failure of an in-feed of the redundant power supply must not cause a failure of the TELEPERM XS cabinets. If there is evidence that the test has been carried out in the manufacturer's inspection, it is not necessary to repeat it in the test field,

- The error spread blockings must meet the requirement specifications by using the status processing, and

- Faults must be correctly alarmed according to the specified alarm concept.

Hardware Failures testing supplements Application Software Functional tests and the Cabinet Alarm Monitoring Tests so that sufficient overlap is provided to ensure the effects due to Hardware Failures are consistent with postulated failure modes (e.g., FMEA) and are handled as required. Testing is performed by simulating the failures at the signal input (i.e., I/O Modules, Optical Communication Interfaces, etc.), and verifying that the effects are consistent with system requirements. Testing also encompasses I/O Failures, Communication Failures, and Loss of Power Failures (i.e., breaker failures) as tested failures. Testing will validate detectable failure modes are in fact detectable. Testing will also verify the appropriate response at the system and sub-system levels. These tests cover mainly the effects of communication failures on the signaling to Gateway and in the control room.

13.3.2.4 Graphic Service Monitor Tests

This testing will verify that the proper connections are made between the Application Software and the project-specific GSM Screens that operate on the TELEPERM XS Service Unit. The signals from each TELEPERM XS processor to the GSM Screens will be manipulated and the values of the signals will be visually verified on the GSM Screens themselves. The signals that are written from the GSM Screens will be manipulated on the GSM Screens and verified on the Service Unit through the use of the project-independent portion of the GSM and/or the dynamic Function Diagram Editor.

Draft

13.3.2.5 Gateway Tests

The purpose of this test is to verify the correct connection to the TELEPERM XS Gateway and the correct functionality of the Application Software implemented on the TELEPERM XS Gateway. This test also verifies that the data link between the TELEPERM XS Gateway and the on-line system can support the specified update rate for all parameters. To perform this test, the Gateway signals from each TELEPERM XS processor will be manipulated and the values of the signals will be validated on the Gateway.

13.3.2.6 System Tests

The purpose of these tests is to demonstrate the ability of the TELEPERM XS System to function during selected plant event scenarios, if required. These tests will demonstrate when the combination of inputs corresponding to the plant event scenario is applied to the TELEPERM XS System, the appropriate system response occurs. Selected test cases will also be specified to validate I&C functions which are spread out across several TELEPERM XS processors to check the effects of asynchronous processor operation.

13.3.2.7 Response Time Tests

This testing will validate the response time of the TELEPERM XS System. The method for performing response time testing is to initiate a trip from simulated inputs, monitor system trip outputs, and is based upon the slowest output response. This testing includes all hardware and software items that make up each trip path in the TELEPERM XS System. The test will verify the response time of each trip function of the TELEPERM XS System on a channel by channel basis. The test will be performed by simulating each input to the trip function and monitoring that input as well as monitoring the corresponding trip relay outputs. The time between the change in the input and the change of the output is the response time. That response time will be evaluated against the required response times defined in the customer specifications.

Draft

13.3.2.8 Cyber Security Tests

The purpose of this test is to validate the Cyber Security of the TELEPERM XS System, and demonstrate that requirements for ensuring cyber security protection of the TELEPERM XS System are met. The Cyber Security test covers the following areas: hardware security, software security, and network security. The tests validate that activation of the operation modes (i.e., Parameterization, Test, and Diagnostics) in all TELEPERM XS processors by the Service Unit is possible if there is an appropriate release(s) and not possible if the release(s) is missing.

13.3.2.9 Diverse Actuation System Tests

Any Diverse Actuation Systems are tested to validate the functionality of the system.

13.3.3 System Validation Test Documents

The System Validation Test Plan is prepared for system validation testing. System Validation Test Plan ensures that the system requirements detailed in the SDRD are properly implemented in the system design. The comprehensiveness of the testing effort should ensure that all functionality defined in the SDD is tested, if simulation testing is not used for validation.

The System Validation Testing is performed under the direction of the Verification and Validation Group. The System Validation Test Plans, Specifications, Procedures, and Reports are prepared in accordance with the Software Verification and Validation Plan and 10 CFR Part 50 Appendix B QA requirements.

System Test Specifications will incorporate the Test-Design Specification and Test-Case Specification into a single document and conform to IEEE Std 829-1983 and IEEE Std 1008-1987. Test Procedures are prepared for each Test Case and contain test scripts that implement the test cases defined in the Test Specifications. The Test Procedures will conform to IEEE Std 829-1983 and IEEE Std 1008-1987.

The Verification and Validation Group uses the Software Requirements Traceability Matrix to ensure that system requirements (from the SDRD) have been tested in the test

Draft

field. The Verification and Validation Group also uses the Software Requirements Traceability Matrix to ensure that software requirements (from the SDD) that have not been tested in the simulation tests are tested in the test field. The Verification and Validation Group independently verifies that the Application Software versions being tested match those listed in the Software Configuration Management documentation.

13.3.4 Test Reporting

The Verification and Validation Group verifies the System Test results (including FAT) by reviewing the Test Item Transmittal Report, Test Log, Test Incident Report, and Test Summary Report to ensure that they demonstrate that the system satisfies the criteria of the System Test Plan and Test Procedures. The Verification and Validation Group verifies the correct versions of software were used in the Acceptance Test.

The Test Item is considered failed if the test script has a syntax error that prevents the script from running or if the test script or the Test Specification is found to be in error (i.e., the results of the test do not match the predicted results described in the Test Specification).

Any errors encountered while performing the test will be documented in the Test Log and Test Incident Report.

The suspension criteria and resumption requirements used for software validation testing are:

- If a discrepancy is found during test execution, the error is documented in the Test Log and the Test Incident Report and, if warranted, the testing resumes.
- A disposition of the discrepancies logged will determine if the discrepancy affects the Test Specification, Test Procedures, Software Requirements Specification, or the project-specific Application Software.

Draft

- If a discrepancy is found while comparing the plot data to the expected results, the discrepancy is recorded, evaluated, and resolved. The discrepancy is recorded in the Test Incident Report and the review of test results continues.

- When a discrepancy is detected that affects the affected design documents or the project-specific Application Software, an Open Item is created and corrected.

- During review of the test results, all discrepancies are recorded in the Test Incident Report.

- Test reruns may start after required changes to the affected design documents and project-specific Application Software have been implemented and the Test Specifications and Test Procedures have been updated to the new design.

- Test reruns are performed on all sections of the Test Specification determined necessary and recorded in the Test Incident Report.

The pass/fail criteria used for system/software validation testing are:

- The Test Item is considered successfully passed when the results of the test match the predicted results described in the Test Specification with no unexpected intermediate results.

- A test Item containing unexpected results may be acceptable considered successfully passed if the evaluation of the unexpected result concludes that the TELEPERM XS Application Software is functioning correctly. Disposition/justification of the item is documented and preserved in the Test Incident Report. Under these conditions, a retest of the item will not be necessary.

- The Test Item is considered failed if the test script has a syntax error that prevents the script from running or if the test script or the Test Specification is

Draft

found to be in error (i.e., the results of the test do not match the predicted results described in the Test Specification).

The Open Items system, as described in Section 3.4, is used to document any discrepancies identified during software validation testing. The Verification and Validation Report lists any verification and validation discrepancies or problems discovered during the software validation tests, and associated anomaly evaluations, in accordance with Section 3.4.

13.3.5 Summary of Test Field Validation Testing

The project-specific test field validation tests satisfy IEEE Std 1012-1998 system and acceptance testing requirements for the project-specific TELEPERM XS systems. This testing always includes test cases to address the limitations of any simulation testing performed as validation testing.

The FAT prerequisite tests (or manufacturing tests) provide integration testing of the TELEPERM XS System hardware components, which addresses one limitation of the simulation environment (i.e., hardware interface). Various system tests are performed during FAT to address the limitations of the simulation environment (i.e., dynamic effects associated with hardware interfaces).

Additional Application Software integration and functional test cases to validate engineering I&C functionality are added to the scope of system testing for the case where simulation testing is not used to satisfy IEEE Std 1012-1998 validation requirements for Application Software validation testing.

Draft

14.0 CONCLUSIONS

The Software Program Manual described the program measures incorporated at AREVA NP to:

- Ensure that the TELEPERM XS Application Software attains a level of quality commensurate with its importance to safety functions
- Ensure that the Application Software performs the required safety functions correctly
- Conform to established technical and documentation requirements, conventions, rules, and industry standards

The Software Program Manual described the plans to address:

- Software Management
- Software Development
- Software Quality Assurance
- Software Integration
- Software Installation
- Software Maintenance and Operations
- Software Training
- Software Safety Analysis
- Software Verification and Validation
- Software Configuration Management
- Software Test Plan

AREVA NP Operating Instructions will be used to define specific implementations details. For example, the Software Configuration Management Plan is defined in an Operating Instruction and additional administrative controls for the software library are

Draft

specified in a separate Operating Instruction. Operating instructions established for these five plans are available onsite at AREVA NP facilities to support NRC review of this topical report.

AREVA NP requests that the NRC issue a Safety Evaluation Report that approves the use of the Software Program Manual described in the topical report. AREVA NP intends to use the Software Program Manual to support digital safety instrumentation and control (I&C) system upgrades at operating nuclear plants and digital safety systems for new nuclear plants. For instance, the approved version of this topical report would be referenced in the Design Control Document for the U.S. EPR.

Draft

15.0 REFERENCES

15.1 U.S. Regulations

1. 10 CFR Part 21, “Reporting of Defects and Noncompliance.”
2. 10 CFR Part 50, Appendix B, “Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants.”

15.2 U.S. Regulatory Guidance

3. Regulatory Guide 1.53, Revision 2, November 2003, “Application of the Single-Failure Criterion to Safety Systems.”
4. Regulatory Guide 1.152, Revision 2, January 2006, “Criteria for Digital Computers in Safety Systems of Nuclear Power Plants.”
5. Regulatory Guide 1.168, Revision 1, February 2004, “Verification, Validation, Reviews, And Audits For Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”
6. Regulatory Guide 1.169, September 1997, “Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”
7. Regulatory Guide 1.170, September 1997, “Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”
8. Regulatory Guide 1.171, September 1997, “Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”
9. Regulatory Guide 1.172, September 1997, “Software Requirements Specifications for Digital Computer Software Used In Safety Systems of Nuclear Power Plants.”
10. Regulatory Guide 1.173, September 1997, “Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”
11. NUREG-0800, Standard Review Plan (SRP), Chapter 7, Branch Technical Position (BTP) 7-14, “Guidance on Software Review for Digital Computer-Based Instrumentation and Control Systems.”

Draft

12. NUREG-0800, SRP, Chapter 7, BTP 7-19, "Guidance for Evaluation of Diversity and Defense-in-Depth in Digital Computer-Based Instrumentation and Control Systems."

15.3 U.S. Industry Standards

13. ANSI/ASME NQA-1 – 1994, "Quality Assurance Program Requirements for Nuclear Facilities."
14. IEEE Std 7-4.3.2-2003, "Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations."
15. IEEE Std 379-2000, "Application of the Single-Failure Criterion to Nuclear Power Generating Station Safety Systems."
16. IEEE Std 352-1987, "Guide for General Principles of Reliability Analysis of Nuclear Power Generating Station Safety Systems."
17. IEEE Std 577-1976, "Requirements for Reliability Analysis in the Design and Operation of Safety Systems for Nuclear Power Generating Stations."
18. IEEE Std 603-1991, "Criteria for Safety Systems in Nuclear Power Plants."
19. IEEE Std 610.12-1990, "Software Engineering Terminology."
20. IEEE Std 730-2002, "Standard for Software Quality Assurance Plans."
21. IEEE Std 828-1990, "Standard for Software Configuration Management Plans."
22. IEEE Std 829-1983, "Standard for Software Test Documentation."
23. IEEE Std 830-1993, "Recommended Practice for Software Requirements Specifications."
24. IEEE Std 1008-1987, "IEEE Standard for Software Unit Testing."
25. IEEE Std 1012-1998, "Standard for Software Verification and Validation."
26. IEEE Std 1028-1997, "Standard for Software Reviews."
27. IEEE Std 1042-1987, "Guide to Software Configuration Management."
28. IEEE Std 1063-2001, "IEEE Standard for User Documentation."
29. IEEE Std 1074-1995, "IEEE Standard for Developing Software Life Cycle Processes."

Draft

30. IEEE Std 1074-1997, "IEEE Standard for Developing Software Life Cycle Processes."

31. IEEE Std 1228-1994, "IEEE Standard for Software Safety Plans."

15.4 Regulatory Review Precedent

32. NRC Safety Evaluation Report for Siemens Topical Report EMF-2110(NP), Revision 1, "TELEPERM XS: A Digital Reactor Protection System," May 5, 2000.

15.5 AREVA NP Documents

33. AREVA NP Records Management Program Manual 1E1, Revision 20.

34. Siemens Topical Report EMF-2110, Revision 1, "TELEPERM XS: A Digital Reactor Protection System," September 1, 1999.

35. AREVA NP Topical Report ANP-10266A, Revision 01, "AREVA NP Inc. Quality Assurance Plan (QAP) for Design Certification of the U.S. EPR."

37. AREVA NP Document No. 56-5015885, "Quality Management Manual."

38. AREVA NP Document No. 105-5017274, "Projects Manual."

39. AREVA NP Project Management Guideline PMG-7, "Risk Management."

40. AREVA NP Administrative Procedure 0405-22, "Design Review Boards."

41. AREVA NP Administrative Procedure 0503-21, "Transmittal of Deliverable Safety-Related Product Documentation to Customers."

42. AREVA NP Administrative Procedure 1702-22, "Employee Training."

43. AREVA NP Administrative Procedure 1707-01, "Evaluation and Reporting of Safety Significant Issues."

44. AREVA NP Administrative Procedure 1717-06, "Corrective Action Program (WebCAP)."

Draft

48. Letter, Ronnie L. Gardner (AREVA NP Inc.) to Document Control Desk (NRC), "Follow-up Actions from December 19, 2007, NRC Audit of ANP-10272, 'Software Program Manual for TELEPERM XS™ Safety Systems Topical Report,' Program Implementation (TAC No. MD3971)," NRC: 08:008, January 24, 2008.

Draft

Appendix A - Map of AREVA NP Software Life Cycle Activities to IEEE Std 1074 Software Life Cycle Model

	AREVA NP Application Life Cycle Activities →	I.A, Evaluate Consistency of Requirements	I.B, Evaluate Completeness of Requirements	I.C Evaluate Requested Schedule vs. Schedule Estimate	I.D, Calculation of Costs	I.E, Evaluate Scope of Supply	I.F, Evaluate Necessary Resources	II.A, Define Work Breakdown Structure	II.B, Project Schedule	II.C, Detailed Calculation of Cost	II.D, Resources Specified	II.E, Create Project Plan	II.F, Project Kickoff	III.A, Create Overall I&C Design Concept	III.B, Create Hardware Design Solutions	III.C, Create Software Design Solutions	III.D, Create Documentation Concept	III.E, Create ID Coding Concept	III.F, Create O&M Concept	III.G, Create Software Life Cycle	III.H, Create Periodic Test Concept
IEEE Std 1074-1995 Software Life Cycle ↓	IEEE Std 1074-1997 Software Life Cycle ↓																				
2.3, Identify Candidate Software Life Cycle Models; 2.4, Select Project Model; 3.1.3, Map Activities to Software Life Cycle Model	A.1.1.1, Create SLCP												X							X	
	A.1.1.2, Perform Estimations		X	X	X	X		X													
3.1.4, Allocate Project Resources	A.1.1.3, Allocate Project Resources						X		X	X	X	X									
3.3.4, Define Metrics	A.1.1.4, Define Metrics																				
3.1.5, Establish Project Environment	A.1.2.1, Plan Evaluations							X	X	X	X	X	X								
7.2.3, Plan Configuration Management	A.1.2.2, Plan Configuration Management							X	X	X	X		X					X			
4.1.6, Plan System Transition	A.1.2.3, Plan System Transition							X	X	X	X		X						X		
6.1.3, Plan Installation	A.1.2.4, Plan Installation							X	X	X	X		X								
7.3.3, Plan Documentation	A.1.2.5, Plan Documentation							X	X	X	X		X				X	X			
7.4.3, Plan the Training Program	A.1.2.6, Plan Training							X	X	X	X		X								
3.1.6, Plan Project Management	A.1.2.7, Plan Project Management							X	X	X	X		X								
5.3.7, Plan Integration	A.1.2.8, Plan Integration							X	X	X	X		X								
	A.1.3.1, Input Information	X	X	X	X													X			
3.2.5, Manage the Project	A.1.3.2, Manage the Project																				
3.3.6, Identify Quality Improvement Needs	A.1.3.3, Identify SLCP Improvement Needs	X																		X	
3.2.6, Retain Records	A.1.3.4, Retain Records																X				
	A.1.3.5, Collect and Analyze Metric Data																				
4.1.3, Identify Ideas or Needs	A.2.1.1 Identify Ideas or Needs																X	X			
4.1.4, Formulate Potential Approaches	A.2.1.2, Formulate Potential Approaches													X	X						
4.1.5, Conduct Feasibility Studies	A.2.1.3, Conduct Feasibility Studies													X	X						
4.1.7, Refine and Finalize the Idea or Need	A.2.1.4, Refine and Finalize the Idea or Need	X	X	X	X									X	X						
4.2.3, Analyze Functions	A.2.2.1, Analyze Functions															X					
4.2.4, Develop System Architecture	A.2.2.2, Develop System Architecture																				
4.2.5, Decompose System Requirements	A.2.2.3, Decompose System Requirements																				
	A.2.3.1, Identify Imported Software Requirements															X					
	A.2.3.2, Evaluate Software Import Sources (If Applicable)															X					
	A.2.3.3, Define Software Import Method (If															X					
	A.2.3.4, Import Software (If Applicable)																				
5.1.3, Define and Develop Software Requirements	A.3.1.1, Define and Develop Software Requirements																	X			

Appendix A - Map of AREVA NP Software Life Cycle Activities to IEEE Std 1074 Software Life Cycle Model

	AREVA NP Application Life Cycle Activities →	III.I, Create Service Concept	IV.A, Create SRS	IV.B Create SDD	IV.C, Create Code	IV.D, Create Test Plan (SIVAT)	IV.E, Test Execution (SIVAT)	IV.F, Test Report (SIVAT)	V.1, Mechanical Hardware Design	V.2, External Interface Design with Wiring Diagrams	VI.A, Test Plan (FAT)	VI.B, Test Procedures (FAT)	VI.C, Test Execution (FAT)	VII.A, Installation and Commissioning Test Plan - Site Acceptance Test	VII.B, Test Procedure - Site Acceptance Test	VII.C, Test Report - Site Acceptance Test	VIII.A, Final Documentation	Notes
IEEE Std 1074-1995 Software Life Cycle ↓	IEEE Std 1074-1997 Software Life Cycle ↓																	
2.3, Identify Candidate Software Life Cycle Models; 2.4, Select Project Model; 3.1.3, Map Activities to Software Life Cycle Model	A.1.1.1, Create SLCP																	
	A.1.1.2, Perform Estimations																	
3.1.4, Allocate Project Resources	A.1.1.3, Allocate Project Resources																	
3.3.4, Define Metrics	A.1.1.4, Define Metrics																	See Software Quality Assurance Plan and Software Verification and Validation Plan.
3.1.5, Establish Project Environment	A.1.2.1, Plan Evaluations																	
7.2.3, Plan Configuration Management	A.1.2.2, Plan Configuration Management																	
4.1.6, Plan System Transition	A.1.2.3, Plan System Transition																	
6.1.3, Plan Installation	A.1.2.4, Plan Installation																	
7.3.3, Plan Documentation	A.1.2.5, Plan Documentation																	
7.4.3, Plan the Training Program	A.1.2.6, Plan Training																	
3.1.6, Plan Project Management	A.1.2.7, Plan Project Management																	
5.3.7, Plan Integration	A.1.2.8, Plan Integration																	
	A.1.3.1, Input Information		X															
3.2.5, Manage the Project	A.1.3.2, Manage the Project																	Project Specific, included in the Project Plan.
3.3.6, Identify Quality Improvement Needs	A.1.3.3, Identify SLCP Improvement Needs																	
3.2.6, Retain Records	A.1.3.4, Retain Records																	
	A.1.3.5, Collect and Analyze Metric Data																	Project Specific, included in the Project Plan.
4.1.3, Identify Ideas or Needs	A.2.1.1 Identify Ideas or Needs		X															
4.1.4, Formulate Potential Approaches	A.2.1.2, Formulate Potential Approaches																	
4.1.5, Conduct Feasibility Studies	A.2.1.3, Conduct Feasibility Studies																	
4.1.7, Refine and Finalize the Idea or Need	A.2.1.4, Refine and Finalize the Idea or Need																	
4.2.3, Analyze Functions	A.2.2.1, Analyze Functions		X															
4.2.4, Develop System Architecture	A.2.2.2, Develop System Architecture																	System Architecture is already defined for TELEPERM XS.
4.2.5, Decompose System Requirements	A.2.2.3, Decompose System Requirements		X															
	A.2.3.1, Identify Imported Software Requirements																	
	A.2.3.2, Evaluate Software Import Sources (If Applicable)																	
	A.2.3.3, Define Software Import Method (If																	
	A.2.3.4, Import Software (If Applicable)																	Controlled by AREVA NP procurement processes.
5.1.3, Define and Develop Software Requirements	A.3.1.1, Define and Develop Software Requirements		X															

Appendix A - Map of AREVA NP Software Life Cycle Activities to IEEE Std 1074 Software Life Cycle Model

	AREVA NP Application Life Cycle Activities →	I.A, Evaluate Consistency of Requirements	I.B, Evaluate Completeness of Requirements	I.C Evaluate Requested Schedule vs. Schedule Estimate	I.D, Calculation of Costs	I.E, Evaluate Scope of Supply	I.F, Evaluate Necessary Resources	II.A, Define Work Breakdown Structure	II.B, Project Schedule	II.C, Detailed Calculation of Cost	II.D, Resources Specified	II.E, Create Project Plan	II.F, Project Kickoff	III.A, Create Overall I&C Design Concept	III.B, Create Hardware Design Solutions	III.C, Create Software Design Solutions	III.D, Create Documentation Concept	III.E, Create ID Coding Concept	III.F, Create O&M Concept	III.G, Create Software Life Cycle	III.H, Create Periodic Test Concept
IEEE Std 1074-1995 Software Life Cycle ↓	IEEE Std 1074-1997 Software Life Cycle ↓																				
5.1.4, Define Interface Requirements	A.3.1.2, Define Interface Requirements																	X			
5.1.5, Prioritize and Integrate Software Requirements	A.3.1.3, Prioritize and Integrate Software Requirements																				
5.2.3, Perform Architectural Design	A.3.2.1, Perform Architectural Design																				
5.2.4, Design Data Base (If Applicable)	A.3.2.2, Design Data Base (If Applicable)																				
5.2.5, Design Interfaces	A.3.2.3, Design Interfaces																				
5.2.7, Perform Detailed Design	A.3.2.4, Perform Detailed Design																				
5.3.4, Create Source Code; 5.3.5, Create Object Code	A.3.3.1, Create Executable Code																				
5.3.6, Create Operating Documentation	A.3.3.2, Create Operating Documentation																		X		
5.3.8, Perform Integration	A.3.3.3 Perform Integration																				
6.1.4, Distribute Software	A.4.1.1, Distribute Software																				
6.1.5, Install Software	A.4.1.2, Install Software																				
6.1.6, Accept Software in Operational Environment	A.4.1.3, Accept Software in Operational Environment																		X		
6.2.3, Operate the System	A.4.2.1, Operate the System																		X		
6.2.4 Provide Technical Assistance and Consulting	A.4.2.2, Provide Technical Assistance and Consulting																		X		
6.2.5, Maintain Support Request Log	A.4.2.3, Maintain Support Request Log																		X		
3.3.6, Identify Quality Improvement Needs	A.4.3.1, Identify Software Improvement Needs																		X	X	
3.2.7, Implement Problem Reporting Methods	A.4.3.2, Implement Problem Reporting Methods																				
6.3.3, Reapply SLC	A.4.3.3, Reapply SLC																				
6.4.3, Notify User	A.4.4.1, Notify User	Incorporated into Software Operations and Maintenance Plan																			
6.4.4, Conduct Parallel Operations (If Applicable)	A.4.4.2, Conduct Parallel Operations (If Applicable)	Not Applicable																			
6.4.5, Retire System	A.4.4.3, Retire System	Incorporated into Software Operations and Maintenance Plan																			
	A.5.1.1, Conduct Reviews	Incorporated into Software Quality Assurance Plan																			
	A.5.1.2, Create Traceability Matrix	Incorporated into Software Verification and Validation Plan																			
	A.5.1.3, Conduct Audits	Incorporated into Software Quality Assurance Plan																			
	A.5.1.4, Develop Test Procedures																				
5.3.3, Create Test Data	A.5.1.5, Create Test Data,																				
	A.5.1.6, Execute Tests																				
	A.5.1.7, Report Evaluation Results																				
7.2.4, Develop Configuration Identification	A.5.2.1, Develop Configuration Identification																				
7.2.5, Perform Configuration Control	A.5.2.2, Perform Configuration Control																				
7.2.6, Perform Status Accounting	A.5.2.3, Perform Status Accounting																				

Appendix A - Map of AREVA NP Software Life Cycle Activities to IEEE Std 1074 Software Life Cycle Model

	AREVA NP Application Life Cycle Activities →	III.I, Create Service Concept	IV.A, Create SRS	IV.B Create SDD	IV.C, Create Code	IV.D, Create Test Plan (SIVAT)	IV.E, Test Execution (SIVAT)	IV.F, Test Report (SIVAT)	V.1, Mechanical Hardware Design	V.2, External Interface Design with Wiring Diagrams	VI.A, Test Plan (FAT)	VI.B, Test Procedures (FAT)	VI.C, Test Execution (FAT)	VII.A, Installation and Commissioning Test Plan - Site Acceptance Test	VII.B, Test Procedure - Site Acceptance Test	VII.C, Test Report - Site Acceptance Test	VIII.A, Final Documentation	Notes
IEEE Std 1074-1995 Software Life Cycle ↓	IEEE Std 1074-1997 Software Life Cycle ↓																	
5.1.4, Define Interface Requirements	A.3.1.2, Define Interface Requirements		X															
5.1.5, Prioritize and Integrate Software Requirements	A.3.1.3, Prioritize and Integrate Software Requirements																	Not required for automatically generated software tool.
5.2.3, Perform Architectural Design	A.3.2.1, Perform Architectural Design																	System Architecture is already defined for TELEPERM XS.
5.2.4, Design Data Base (If Applicable)	A.3.2.2, Design Data Base (If Applicable)																	
5.2.5, Design Interfaces	A.3.2.3, Design Interfaces			X														
5.2.7, Perform Detailed Design	A.3.2.4, Perform Detailed Design			X														
5.3.4, Create Source Code; 5.3.5, Create Object Code	A.3.3.1, Create Executable Code				X													
5.3.6, Create Operating Documentation	A.3.3.2, Create Operating Documentation																	
5.3.8, Perform Integration	A.3.3.3 Perform Integration				X													
6.1.4, Distribute Software	A.4.1.1, Distribute Software				X													
6.1.5, Install Software	A.4.1.2, Install Software				X													
6.1.6, Accept Software in Operational Environment	A.4.1.3, Accept Software in Operational Environment				X													
6.2.3, Operate the System	A.4.2.1, Operate the System				X													
6.2.4 Provide Technical Assistance and Consulting	A.4.2.2, Provide Technical Assistance and Consulting	X																
6.2.5, Maintain Support Request Log	A.4.2.3, Maintain Support Request Log	X																
3.3.6, Identify Quality Improvement Needs	A.4.3.1, Identify Software Improvement Needs	X																
3.2.7, Implement Problem Reporting Methods	A.4.3.2, Implement Problem Reporting Methods	X																
6.3.3, Reapply SLC	A.4.3.3, Reapply SLC	X																
6.4.3, Notify User	A.4.4.1, Notify User																	
6.4.4, Conduct Parallel Operations (If Applicable)	A.4.4.2, Conduct Parallel Operations (If Applicable)																	
6.4.5, Retire System	A.4.4.3, Retire System																	
	A.5.1.1, Conduct Reviews																	
	A.5.1.2, Create Traceability Matrix																	
	A.5.1.3, Conduct Audits																	
	A.5.1.4, Develop Test Procedures					X												
5.3.3, Create Test Data	A.5.1.5, Create Test Data,					X					X	X		X	X			
	A.5.1.6, Execute Tests						X						X		X			
	A.5.1.7, Report Evaluation Results							X	X	X						X		
7.2.4, Develop Configuration Identification	A.5.2.1, Develop Configuration Identification																X	
7.2.5, Perform Configuration Control	A.5.2.2, Perform Configuration Control																X	
7.2.6, Perform Status Accounting	A.5.2.3, Perform Status Accounting																X	

	AREVA NP Application Life Cycle Activities →	I.A, Evaluate Consistency of Requirements	I.B, Evaluate Completeness of Requirements	I.C Evaluate Requested Schedule vs. Schedule Estimate	I.D, Calculation of Costs	I.E, Evaluate Scope of Supply	I.F, Evaluate Necessary Resources	II.A, Define Work Breakdown Structure	II.B, Project Schedule	II.C, Detailed Calculation of Cost	II.D, Resources Specified	II.E, Create Project Plan	II.F, Project Kickoff	III.A, Create Overall I&C Design Concept	III.B, Create Hardware Design Solutions	III.C, Create Software Design Solutions	III.D, Create Documentation Concept	III.E, Create ID Coding Concept	III.F, Create O&M Concept	III.G, Create Software Life Cycle	III.H, Create Periodic Test Concept
IEEE Std 1074-1995 Software Life Cycle ↓	IEEE Std 1074-1997 Software Life Cycle ↓																				
7.3.4, Implement Documentation	A.5.3.1, Implement Documentation																				
7.3.5, Produce and Distribute Documentation	A.5.3.2, Produce and Distribute Documentation																				
7.4.4, Develop Training Materials	A.5.4.1, Develop Training Materials																				
7.4.5, Validate the Training Program	A.5.4.2, Validate the Training Program																				
7.4.6, Implement the Training Program	A.5.4.3, Implement the Training Program																				
3.2.3, Analyze Risks																					
3.2.4, Perform Contingency Planning																					
3.3.3, Plan Software Quality Management																					
3.3.5, Manage Software Quality																					
5.2.6, Design or Select Algorithms																					
7.1.3, Plan Verification and Validation																					
7.1.4, Execute Verification and Validation Tasks																					
7.1.5, Collect and Analyze Metric Data																					
7.1.6 Plan Testing (Verification and Validation)																					
7.1.7, Develop Test Requirements (Verification and Validation)																					
7.1.8, Execute the Tests (Verification and Validation)																					

	AREVA NP Application Life Cycle Activities →	III.I, Create Service Concept	IV.A, Create SRS	IV.B Create SDD	IV.C, Create Code	IV.D, Create Test Plan (SIVAT)	IV.E, Test Execution (SIVAT)	IV.F, Test Report (SIVAT)	V.1, Mechanical Hardware Design	V.2, External Interface Design with Wiring Diagrams	VI.A, Test Plan (FAT)	VI.B, Test Procedures (FAT)	VI.C, Test Execution (FAT)	VII.A, Installation and Commissioning Test Plan - Site Acceptance Test	VII.B, Test Procedure - Site Acceptance Test	VII.C, Test Report - Site Acceptance Test	VIII.A, Final Documentation	Notes
IEEE Std 1074-1995 Software Life Cycle ↓	IEEE Std 1074-1997 Software Life Cycle ↓																	
7.3.4, Implement Documentation	A.5.3.1, Implement Documentation																X	
7.3.5, Produce and Distribute Documentation	A.5.3.2, Produce and Distribute Documentation																X	
7.4.4, Develop Training Materials	A.5.4.1, Develop Training Materials																	Project Specific, included in the Project Plan.
7.4.5, Validate the Training Program	A.5.4.2, Validate the Training Program																	Project Specific, included in the Project Plan
7.4.6, Implement the Training Program	A.5.4.3, Implement the Training Program																	Project Specific, included in the Project Plan.
3.2.3, Analyze Risks																		Part of the generic AREVA NP Project Management Manual
3.2.4, Perform Contingency Planning																		Part of the generic AREVA NP Project Management Manual.
3.3.3, Plan Software Quality Management																		Covered in Software Program Manual and Software Quality Assurance Plan.
3.3.5, Manage Software Quality																		Covered in Software Program Manual and Software Quality Assurance Plan
5.2.6, Design or Select Algorithms																		Algorithms selected by the customer.
7.1.3, Plan Verification and Validation																		Addressed in Software Verification and Validation Plan.
7.1.4, Execute Verification and Validation Tasks																		Addressed in Software Verification and Validation Plan.
7.1.5, Collect and Analyze Metric Data																		Addressed in Software Verification and Validation Plan.
7.1.6 Plan Testing (Verification and Validation)																		Addressed in Software Verification and Validation Plan.
7.1.7, Develop Test Requirements (Verification and Validation)																		Addressed in Software Verification and Validation Plan.
7.1.8, Execute the Tests (Verification and Validation)																		Addressed in Software Verification and Validation Plan.

Appendix B – Historical Information on SPACE Tool Qualification

The TELEPERM XS system is a fully integrated suite of hardware and software designed specifically for nuclear safety applications. The TELEPERM XS system has significant nuclear operating experience. The TELEPERM XS platform has been fully qualified as an integrated platform. The TELEPERM XS system is described in TELEPERM XS Topical Report. NRC approved the TELEPERM XS Topical Report in a safety evaluation report issued in May 2000.

B.1 TELEPERM XS Qualification Process

The overall qualification process for the TELEPERM XS system is shown in Figure 3.1. The qualification process is a two-part process: generic system qualification and specific system qualification. The qualification process for Application Software starts with the application-independent (generic) qualification process described in Section 2.1 of the TELEPERM XS Topical Report. The generic qualification process included an integration and system test phase. The specific system used for this generic qualification step is described in detail in Section 3.2.2 of the TELEPERM XS Topical Report.

The overall application independent qualification process is described in Section 2.2 of the TELEPERM XS Topical Report. The TELEPERM XS platform qualification process is shown in Figure 3-1.

The generic qualification of the Application Software development process includes work performed by AREVA NP (GmbH) and qualification work performed by an independent third party. Section 2.4.3.3.2 of the TELEPERM XS Topical Report describes part of the AREVA NP activities. Section 2.4.3.3.3 of the TELEPERM XS Topical Report describes part of the third party activities. The application-dependent (specific project) phase takes credit for all application-independent (generic) qualification activities, as noted on page 2-4 of the TELEPERM XS Topical Report. The

Draft

Software Program Manual was prepared to describe the specific system qualification process for TELEPERM XS projects in the U.S.

B.2 TELEPERM XS Project-Specific Activities

The TELEPERM XS Topical Report described a general framework for the implementation of individual projects using the TELEPERM XS technology in Section 5.1.3. An integral part of that framework is the use of the TELEPERM XS engineering tools set for the development of Application Software. A key feature of the TELEPERM XS tool set is the safety-related automatic code generator in the SPACE tool, which was approved as part of the TELEPERM XS Topical Report. The development of TELEPERM XS Application Software is predicated on the use of the SPACE tool for code generation.

The TELEPERM XS Topical Report describes the use of a simulator-based validation process for TELEPERM XS Application Software in Section 2.4.3.3.2. The role of the simulator validation tool in the standard AREVA NP engineering process for TELEPERM XS project implementation is shown in TELEPERM XS Topical Report Figure 2.8. The correctness of TELEPERM XS code generation in the course of application projects is covered by validation activities (i.e., simulation and/or test field environments). RETRANS analysis is not considered to be part of the standard TELEPERM XS engineering process for Application Software.

The application-dependent (specific project) phase takes credit for all application-independent (generic) qualification activities, as noted on page 2-4 of the TELEPERM XS Topical Report. The Software Program Manual describes the specific system qualification process for TELEPERM XS projects in the U.S. This process uses the SIVAT tool, as described below.

Application software is developed using the TELEPERM XS Specification and Coding Environment (SPACE) tool. This tool is used to develop Function Diagrams and Network Diagrams. Function Diagrams specify the signal processing requirements for

Draft

the system. Network Diagrams define the hardware components of the system and their logical interconnections. Software code is automatically generated from the Function Diagrams and Network Diagrams by the SPACE tool. Logical 'software integration' occurs at this stage. The project-specific TELEPERM XS system is developed from qualified hardware and software modules using the qualified development tools.

Physical software integration occurs during the FAT stage, when the Application Software is loaded on the TELEPERM XS processors. The project-specific System Test Plan covers the approach and activities associated with the Software and Hardware Integration.

A project-specific Software Generation and Download Procedure is issued for each project to control and document the generation of each Application Software release. It is used to control and document the download of each approved software release to the target system. This project-specific Software Generation and Download Procedure is implemented under a work order (task-letter) for each Application Software Release. The Software Generation and Download Procedure is a configuration item that is governed by the Software Configuration Management Plan.

B.3 Validation of SPACE Tool Automatic Code Generator

The TELEPERM XS Topical Report Section 2.4.3.3.3 (in addition to Section 3.2.1) described generic qualification activities for the TELEPERM XS SPACE tool automatic code generator, which included the development of an independent code verification tool for checking code generator output and tool-based checks for generated TELEPERM XS Application Software for the first I&C projects. RETRANS is the independent code verification tool used in the qualification process of the TELEPERM XS automatic code generator in the SPACE Tool.

ISTec (Institute of Safety Technology) is a subsidiary of the German Society for Reactor Safety (GRS). GRS-ISTec has a major role in the TELEPERM XS third party

Draft

assessment of system concepts and safety-related software (generic qualification). RETRANS was developed by GRS-ISTec in order to verify that code generated by means of the SPACE tool code generator complies with the design rules given for the generated Function Diagram and Function Diagram Group modules, and that the functions contained in the generated code are equivalent with the specification data contained in the SPACE database. GRS-ISTec is the RETRANS tool owner and responsible for tool support. In this context, RETRANS is used to validate correct performance of the automatic code generator as part of the generic qualification process.

The TELEPERM XS Topical Report did not specify additional application specific testing using RETRANS as part of the standard TELEPERM XS engineering process. The standard TELEPERM XS engineering methods have not changed.

B.4 RETRANS Tool Capabilities

The RETRANS tool has two capabilities: verification of source code generated by TELEPERM XS automatic code generator and cross-check of TELEPERM XS engineering data specified in redundant trains of the I&C system.

The verification of source code generated by TELEPERM XS code generator is performed by analysis and retranslation of TELEPERM XS application code generated by SPACE tool code generator. The reconstructed specifications from the code retranslation are compared with specification data contained in the TELEPERM XS engineering database (SPACE database). This feature was described in the TELEPERM XS Topical Report for validation of the automatic code generator. This feature of RETRANS is used to revalidate the automatic code generator after changes through the third-party generic qualification process.

The cross-check feature of RETRANS was added to the tool after the NRC Safety Evaluation Report for the TELEPERM XS Topical Report was issued. This feature of the tool is used to detect differences in the functionality of Application Software in the

Draft

redundant divisions of an I&C system. The tool performs an analysis of logics and parameter data specified for redundant system trains and identifies differences in functionality. The differences must be evaluated by an engineer to determine whether the differences are planned (engineered differences) or unplanned (errors). AREVA NP used this feature for a period of time then replaced the capability with a separate tool (***rediff***). AREVA NP developed the ***rediff*** tool for ergonomic and efficiency reasons. The AREVA NP ***rediff*** tool is used for every TELEPERM XS project to support TELEPERM XS Application Software verification tasks. The ***rediff*** tool was designed and implemented using internal QA procedures.

B.5 Experience with RETRANS

As noted in TELEPERM XS Topical Report Section 2.4.3.3.3, AREVA NP (formerly Siemens KWU) used the RETRANS tool as an additional verification of the generically qualified automatic code generators for the first few projects (introductory phase). This approach was taken to gain experience and create a high level of confidence in the automatic code generation tool used for TELEPERM XS Application Software during the introductory phase.

The RETRANS tool was used by AREVA NP on the following projects during the introductory phase:

- Bohunice (Slovakia) in March 2000
- FRM2 (Germany) in April 2000
- KKP (Germany) in May 2000
- PAKS 1 (Hungary) in September 2000
- PAKS 2 (Hungary) in March and September 2000
- PAKS 3 (Hungary) in September 2001

The RETRANS verification relies on the (generically defined) program structure of generated Function Diagram and Function Diagram Group modules and on the data

Draft

model of the SPACE database. Once it has been verified and tested that a code generator version generates code in compliance with this program structure and equivalent with the input data from the SPACE database, RETRANS cannot identify errors in the course of a TELEPERM XS application project implementation. Therefore, RETRANS analysis was not defined to be a step in the TELEPERM XS engineering process.

GRS-ISTec also performs independent testing of Application Software for clients using RETRANS. Code verification projects performed by GRS-ISTec include:

- Bohunice (Slovakia)
- PAKS (Hungary)
- Beznau (Switzerland)
- FRM2 (Germany)

In the very beginning of the TELEPERM XS code generator development (i.e., versions prior to R2.3x) some bugs were identified in the code generator and in RETRANS. These versions of the code generator were prior to deployment of TELEPERM XS systems into nuclear power plants. No findings concerning the SPACE code generator have been identified in later versions.

B.6 Changes to RETRANS Tool

The different RETRANS tool versions are shown in the following table.

RETRANS Tool	Runs with Operating System	For Use with TELEPERM XS Software Version
V2.20	HP-UX 9.0x	R2.2x
V3.0, 3.1, 3.2, 3.3	HP-UX 10.20	R2.33 – 2.38
V4.0 (purchased by NRC)	SUSE LINUX 8.0	R3.0.0 – 3.0.9
V5.0	SUSE LINUX 9.x	R3.1.4 -3.1.5
V6.0 scheduled for 2008	SUSE LINUX 10.x	R3.2/R3.3

Draft

Code analysis is based on well-defined structures of generated TELEPERM XS application code. These structures have not changed since the initial generic qualification of TELEPERM XS. Later developments of RETRANS consisted of adaptations to minor modifications in the SPACE data base structure and adaptations to new operating system versions of the engineering workstation. RETRANS is going to be adapted by GRS-ISTec to the current release of the TELEPERM XS core software in 2008. GRS-ISTec plans to test every modification of the TELEPERM XS code generator presented by AREVA NP using RETRANS.

Draft