Security-Related Information Withheld - Public Version

AREVA

ANP-10272
Revision 2

Software Program Manual for TELEPERM XS™ Safety Systems

Topical Report

April 2010

AREVA NP Inc.

Non-Proprietary

(c) 2010 AREVA NP Inc.

Copyright © 2010

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

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.

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:

- Software Management Plan, which describes the overall management process used for the development of project-specific TELEPERM XS Application Software.
- Software Development Plan, which describes the life cycle activities for TELEPERM XS Application Software development.
- 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.
- Software Integration Plan, which describes the software integration process and the hardware/software integration process for TELEPERM XS projects.
- Software Installation Plan, which describes the installation process for TELEPERM XS projects.
- Software Operations and Maintenance Plan, which describes post-customer delivery TELEPERM XS Software practices.

- 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.
- 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.
- Software Verification and Validation Plan, which describes the method that ensures correctness of the TELEPERM XS Application Software.
- Software Configuration Management Plan, which describes the method that maintains the project-specific TELEPERM XS Software in a controlled configuration.
- 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.

Nature of Changes

Revision	Section(s) or Page(s	Description and Justification
0	All	Initial issue.
1	All	Revision 1 was completely revised and reformatted to address prior RAI responses, alignment with the recent version of Branch Technical Position 7-14, and incorporate new material based on the lessons learned during the review of the Oconee digital protection system license amendment request.
	Section 1.0	Revised to show alignment with NRC Branch Technical Position 7-14 suggested plans.
	Section 2.0	New section that incorporates material from old Section 1.4
	Section 3.0	New section added to address the software management plan. It incorporates material from old Section 1.1, 2.0, and 10.0 and expanded to address prior RAI responses and incorporate new material based on the lessons learned during the review of the Oconee digital protection system license amendment request.
	Section 4.0	New section added to address the software development plan. It incorporates information from old Sections 1.2, 1.3 and 9.0 and expanded to address prior RAI responses and incorporate new material based on the lessons learned during the review of the Oconee digital protection system license amendment request. A new process for evaluating changes to the TELEPERM XS platform was added in Section 4.1.
	Section 5.0	Contains the information from old Section 3.0 and expanded to address prior RAI responses and incorporate new material based on the lessons learned during the review of the Oconee digital protection system license amendment request.
	Section 6.0	New section added to address software integration.
	Section 7.0	New section added to address software integration.
	Section 8.0	Contains the information from old section 7.0.
	Section 9.0	Contains the information from old section 8.0.
	Section 10.0	Contains the information from old section 4.0 and expanded to address prior RAI responses and incorporate new material based on the lessons learned during the review of the Oconee digital protection system license amendment request.

	Section 11.0	Contains the information from old section 6.0 and expanded to address prior RAI responses and incorporate new material based on the lessons learned during the review of the Oconee digital protection system license amendment request.
	Section 12.0	Contains the information from old section 5.0 and expanded to address prior RAI responses and incorporate new material based on the lessons learned during the review of the Oconee digital protection system license amendment request.
	Section 13.0	New section added to address software testing.
	Section 14.0	Contains the information from old section 11.0.
	Section 15.0	Contains the information from old section 12.0 and new references added.
	Appendix A	Revised to reflect changes in Section 4.0.
	Appendix B	Added to incorporate historical information on SPACE Engineering Tool qualification.
2	Section 1.0	Revised to add discussion on how the topical report will be used after approval.
	Section 2.0	New definitions added based on changes to Section 13 and addition of Appendix C. Description of tools deleted based on expanded discussion in Section 4.2.
	Section 3.0	Added discussion of developer software security controls. Deleted of plant-specific action items. Added typical organization chart. Added discussion on resolution of open items.
	Section 4.0	Expanded discussion of software development tools. Expanded discussion of software development lifecycle model. Revised discussion of software testing based on changes to Section 13.
	Section 5.0	Expanded discussion of responsibilities for performance of software quality assurance plan activities. Added new activity to audit change management records. Revised discussion of software testing based on changes to Section 13.
	Section 6.0	Revised discussion of software integration testing based on changes to Section 13.
	Section 7.0	Revised discussion of software installation to reflect vendor scope and order of software installation. Added discussion software installation scripts, maintenance laptop software, and Service Unit software.
	Section 8.0	Revised discussion of software operational and maintenance support to reflect vendor scope
	Section 9.0	Revised discussion of software training support to reflect vendor scope
	Section 10.0	Expanded discussion of code generation process. Revised discussion of software testing based on changes to Section 13.

Section 11.0	Revised discussion of software integration testing based on changes to Section 13. Expanded discussion of verification and validation summary report contents.
Section 12.0	Revised discussion of software configuration control board.
Section 13.0	Revised discussion of software testing to reflect lessons learned from NRC review of the Oconee digital protection system testing.
Section 14.0	Revised to add discussion on how the topical report will be used after approval.
Section 15.0	Added references based on addition of Appendix C.
Appendix A	Revised to simplify presentation and reflect changes in Section 4.0.
Appendix C	Added to discussion on software and data security previous submitted in separate documents.
Appendix D	Relocated TELEPERM XS platform change evaluation process details from Section 4.1.2 and 4.1.3 to new appendix.

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
- **3.2** Organization 3-10
- **3.3** Roles and Responsibilities 3-11
- **3.4** TELEPERM XS Application Software Development Controls 3-17
- **3.5** Problem Reporting 3-18
- **3.6** Use of Industry Standards 3-22

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 4-5
- **4.3** TELEPERM XS Application Software Life Cycle 4-11
- **4.4** Software Classification 4-24
- **4.5** Software Development Documentation 4-27
- **4.6** Security and Disaster Recovery 4-33
- **4.7** Documentation Standards 4-35
- **4.8** User Manuals 4-40

5.0 SOFTWARE QUALITY ASSURANCE PLAN 5-1

- **5.1** Purpose 5-1
- **5.2** Management 5-2
- **5.3** Documentation 5-3
- **5.4** Software Reviews 5-3
- **5.5** Software Audits 5-6
- **5.6** Software Testing 5-8
- **5.7** Standards 5-10
- **5.8** Problem Reporting and Corrective Action 5-11
- **5.9** Tools, Methodologies, and Metrics 5-11
- **5.10** Media Control 5-13
- **5.11** Supplier Control 5-14
- **5.12** Records Collection, Maintenance, and Retention 5-14
- **5.13** Training 5-15
- **5.14** Risk Management 5-15

6.0 SOFTWARE INTEGRATION PLAN 6-1

- **6.1** TELEPERM XS Platform Integration 6-1
- **6.2** TELEPERM XS Application Software Integration 6-2
- **6.3** TELEPERM XS Project Hardware and Software Integration 6-3

7.0 SOFTWARE INSTALLATION PLAN 7-1

- **7.1** TELEPERM XS Software Installation 7-1

7.2 TELEPERM XS Software Installation Verification 7-4
7.3 TELEPERM XS Service Unit Software Installation 7-4
7.4 TELEPERM XS Maintenance Laptop Software Installation 7-5
8.0 SOFTWARE MAINTENANCE AND OPERATIONS PLAN 8-1
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 9-2
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
10.4 Documenting and Correcting Safety Hazards 10-11
11.0 SOFTWARE VERIFICATION AND VALIDATION PLAN 11-1
11.1 Purpose 11-1
11.2 Overview 11-1
11.3 Organization 11-4
11.4 Procedures 11-5
11.5 Schedule 11-6
11.6 Software Integrity Levels 11-6
11.7 Resources 11-7
11.8 Responsibilities 11-7
11.9 Verification and Validation Methods 11-8
11.10 Independent Testing and Validation 11-20
11.11 Regression Testing 11-20
11.12 Metrics 11-21
11.13 Verification and Validation Reports 11-22
12.0 SOFTWARE CONFIGURATION MANAGEMENT PLAN 12-1
12.1 Introduction 12-1
12.2 Management 12-2
12.3 Software Configuration Management Process Controls 12-6
12.4 Software Configuration Management Implementation Activities 12-9
13.0 SOFTWARE TEST PLAN 13-1
13.1 Purpose 13-1

13.2 Alignment with IEEE Std 1012-1998 Testing Activities 13-2
13.3 Application Software Integration Testing 13-6
13.4 Hardware and Software Integration Testing 13-7
13.5 Application Software System and Acceptance Testing 13-9
13.6 Validation Test Documents 13-14
13.7 Test Reporting 13-15
13.8 Test Tools 13-17
13.9 Software Testing Summary 13-18

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 15-4

List of Tables

Page
Table 1-1: AREVA NP Alignment with BTP 7-14 1-3
Table 3-1: List of Industry Standards Used for Application Software Development .. 3-23
Table 4-1: TELEPERM XS Project Input, Tasks, Processes, Output, and Results ... 4-15
Table 4-2: TELEPERM XS Project Phase Correlation 4-22
Table 4-3: Software Integrity Scheme Comparison 4-24
Table 13-1: Alignment with IEEE Std 1012-1998 Test Activities 13-5

List of Figures

Page
Figure 3-1: Annotated Version of Figure 2.2 from 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 3-4: Typical Project Organization Chart 3-11
Figure 4-1: Hierarchical Structure of the Application Software Design 4-38
Figure 11-1: Test Document Work Flow and Control Points 11-3

Appendices

Appendix A - Appendix A - Map of AREVA NP Software Life Cycle Activities to IEEE Std 1074 Software Life Cycle Model
Appendix B - Historical Information on SPACE Engineering Tool Qualification
Appendix C - TELEPERM XS Software and Data Security Features
Appendix D - TELEPERM XS Platform and Procedure Change Evaluation Process

Nomenclature

Acronym	Definition
ASL	Approved Supplier List
BTP	Branch Technical Position
CD-ROM	Compact Disc – Read Only Memory
CFR	Code of Federal Regulations
CPU	Central Processor Unit
CRC	Cyclic Redundancy Checksum
EEPROM	Electrically Erasable Programmable Read-Only Memory
EPRI	Electric Power Research Institute
ERBUS	TELEPERM XS computer-assisted test system for TELEPERM XS test field application (test field simulator)
FAT	Factory Acceptance Test
FEPROM	Flash Erasable Programmable Read-Only Memory
FMEA	Failure Modes and Effects Analysis
GSM	Graphic Service Monitor
I&C	Instrumentation and Control
I/O	Input/Output
ID	Identification
IEEE	Institute of Electrical and Electronic Engineers
ISG	Interim Staff Guidance
IT	Information Technology
MAC	Media Access Code
MIC File	Loadable Code
MSI	Monitoring and Service Interface
NEI	Nuclear Energy Institute
NIDS	Network Intrusion Detection Systems
NIPS	Network Intrusion Protection Systems
NRC	Nuclear Regulatory Commission
NUPIC	Nuclear Procurement Issues Committee
PC	Personal Computer
QA	Quality Assurance
QDS	Qualified Display System

RAM	Random Access Memory
SCM	Software Configuration Management
SDD	Software Design Description
SysRS	System Requirements Specification
SIL	Software Integrity Level
SME	Subject Matter Expert
SPACE	Specification and Coding Environment
SRS	Software Requirements Specification
SysDD	System Description Document
TCP/IP	Transmission Control Protocol/Internet Protocol
V&V	Verification and Validation
WAN	Wide Area Network

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 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 is AREVA NP's upper tier requirements document for TELEPERM XS application software development. The statements in the Software Program Manual are considered to be the programmatic requirements necessary to implement the TELEPERM XS application software program. No specific convention was used to denote requirements. Instead, optional or conditional items are described using the terminology 'may'.

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 Branch Technical Position (BTP) 7-14 (Reference 17) are addressed.

The Software Program Manual will be implemented by procedures to produce the documentation that demonstrates that the following project-specific implementation activities have been successfully accomplished for each life cycle activity group for the following:

- Software Safety Analysis Activities described in Section 10.0,
- Software Verification and Validation Activities described in Section 11.0,
- Software Configuration Management Activities described in Section 12.0, and
- Software Test Activities described in Section 13.0.

These activities are designed to ensure that the:

- Project-specific Application Software is safe,
- System safety requirements have been adequately addressed for each activity group,
- No new hazards have been introduced, and
- Software requirements, design elements, and code elements that can affect safety have been identified, and that all other software requirements, design, and code elements will not adversely affect safety.

The Software Program Manual will be implemented by procedures to produce the project-specific design outputs in the Software Development Plan described in Section 4.0 and produced in accordance with the Software Management Plan described in Section 3.0 and the Software Quality Plan described in Section 5.0.

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 Final

Safety Assessment Report supporting the Design Certification Application for the U.S. EPR.

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	5.0
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

2.0 DEFINITIONS

Acceptance Testing [IEEE Std 610.12 (Reference 28)]

1. Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system.

2. Formal testing conducted to enable a user, customer, or other authorized entity to determine whether to accept a system or component.

Anomaly [IEEE Std 1012 (Reference 34)]

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 Specification and Coding Environment (SPACE) Engineering Tool. The platform system software uses this configuration data to carry out the application specific functionality of the I&C system.

AREVA NP

Designation used in this report to refer to the AREVA NP organization that encompasses the activities of both AREVA NP GmbH and AREVA NP Inc.

AREVA NP GmbH

Designation used in this report to refer to the AREVA NP organization responsible for the TELEPERM XS System development. This organization is based in Erlangen, Germany.

AREVA NP Inc.

The designation AREVA NP Inc., as used in this report, to refer to the AREVA NP organization responsible for the design of U.S. projects using the TELEPERM XS System. This organization is based in Alpharetta, Georgia.

Baseline [IEEE Std 610.12]

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.

Black Box [IEEE Std 610.12]

1. A system or component whose inputs, outputs, and general function are known but whose contents or implementation are unknown or irrelevant. *Contrast with:* white box.

2. Pertaining to an approach that treats a system or component as in 1 above.

CCITT

Comité Consultatif International Téléphonique et Télégraphique, an organization that sets international communications standards.

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.

Component Testing [IEEE Std 610.12]

Testing of individual hardware or software components or groups of related components.

Configuration [IEEE Std 828 (Reference 30)]

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.

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.

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.

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 SysRS or the SRS. These 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.

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.

Functional Testing [IEEE Std 610.12]

1. Testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions. *Syn:* black-box testing. *Contrast with:* structural testing.

2. Testing conducted to evaluate the compliance of a system or component with specified functional requirements.

Integration Testing [IEEE Std 610.12]

Testing in which software components, hardware components, or both are combined and tested to evaluate the interaction between them

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.
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

MD5

Message-Digest algorithm 5 is a widely used cryptographic hash function.

MIC File

Machine language loadable code file.

MicroNET

TELEPERM XS communication software used for message transfers between the processing modules. The communication tasks performed by MicroNET for the processing modules are restricted to providing the communication processors with a description table for the transmission channels on initialization, performing read or write accesses to data in the dual port RAM of the communication processor and performing self-monitoring for communication.

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.

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.

RIPEMD

The RACE Integrity Primitives Evaluation Message Digest system of message digest algorithms and cryptographic hash functions.

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 SysRS for a project.

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.

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 Hazard [Based on IEEE Std 1228]

A software design error that could lead to an unintended operation or failure to operate when required.

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 Risk [Based on IEEE Std 1228 (Reference 40)]

A measure that combines both the likelihood that a software hazard will cause a problem and the severity of that problem.

SPACE

The SPACE Engineering Tool 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
- Software authentication, using **reflist** and **scanmic**
- Software loading
- Load analysis tool
- Database administration

Structural Testing [IEEE Std 610.12]

Testing that takes into account the internal mechanism of a system or component. Types include branch testing, path testing, statement testing. *Syn:* white-box testing. *Contrast with:* functional testing

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 Requirements Specification (SysRS)

A document that specifies, in detail, the requirements of the system to be installed (new or replaced). The SysRS 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.

System Testing [IEEE Std 610.12]

Testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

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.

Test Unit [IEEE Std 1012]

A set of one or more computer program modules together with associated control data, (for example, tables), usage procedures, and operating procedures that satisfy the following conditions:

1. All modules are from a single computer program
2. At least one of the new or changed modules in the set has not completed the unit test
3. The set of modules together with its associated data and procedures are the sole object of a testing process

Unit [IEEE Std 610.12]

1. A separately testable element specified in the design of a computer software component.
2. A logically separable part of a computer program.
3. A software component that is not subdivided into other components.

Unit Testing [IEEE Std 610.12]

Testing of individual hardware or software units or groups of related units.

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.

Watchdog

A device that monitors the cyclic processing in the CPU. It is implemented by a hardware timer on the CPU board.

White Box [IEEE Std 610.12]

1. A system or component whose internal contents or implementation are known.

   Contrast with: black box.

2. Pertaining to an approach that treats a system or component as in 1 above.

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 Service 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.

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 software and data security guidance.

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 60). 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. Audit findings identified by NUPIC are resolved through the AREVA NP Corrective Action Program.

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 59) was issued to describe the Quality Assurance Plan applicable to the Design Certification of the U.S. EPR. The plan meets the requirements of the eighteen point criteria of 10 CFR 50, Appendix B (Reference 3), and ANSI/ASME NQA-1-1994 (Reference 21) 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.

The Software Program Manual describes the program measures incorporated by AREVA NP to ensure that the Application Software attains a high level of quality required for safety-related software, 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 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 the use of Operating Instructions, which implement the requirements described in this Topical Report. Changes to Operating Instructions are evaluated using the process specified in Section 4.1 to ensure that the compliance with the Software Program Manual is maintained.

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 Management System. The record storage requirements are described in the AREVA NP Records Management Program Manual (Reference 54).

3.1.2 Interface with the TELEPERM XS Topical Report

The TELEPERM XS Topical Report (Reference 55) 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 46), 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 complements 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 application-dependent (project-specific) phase takes credit for all application-independent (generic) qualification activities (as noted on page 2-4 of the TELEPERM XS Topical Report). The TELEPERM XS Software Program Manual describes the specific system qualification process for TELEPERM XS projects in the U.S. The relationship of the two Topical Reports is shown in Figure 3-1.

The Software Program Manual describes the standard engineering process used to develop TELEPERM XS Application Software for U.S. projects. The Software Program

Manual is used to address plant-specific action items 2 and 17 from the NRC Safety Evaluation Report for the TELEPERM XS Topical Report. Specifically, action item 2 is addressed by using the configuration management and verification and validation industry standards addressed in Section 3.6 and as described in Sections 11.0 and 12.0. Action item 17 is addressed by the inclusion of the Application Software Requirements Traceability Matrix as a required development process document, as noted in Section 4.5.7.

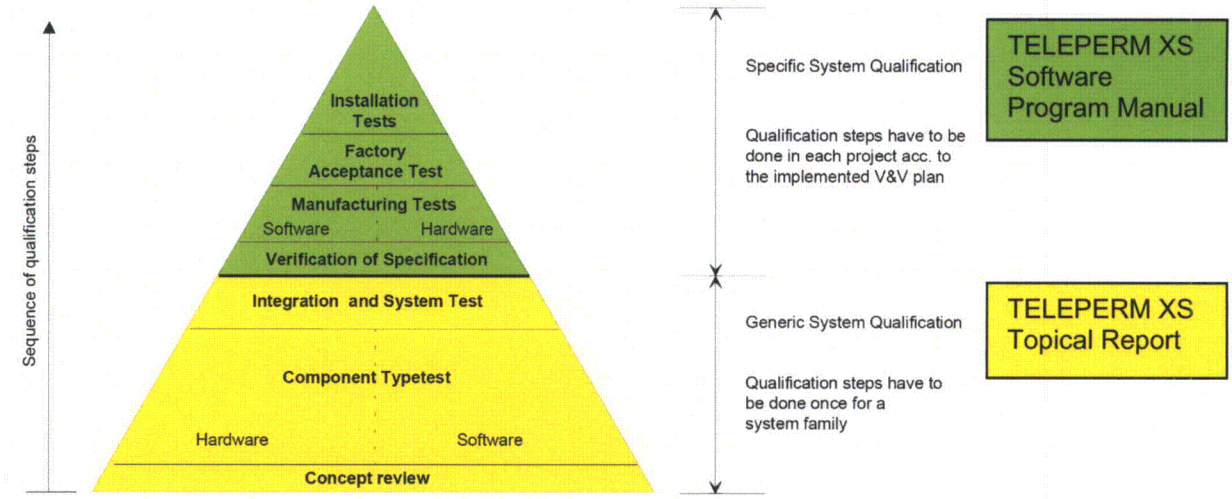


Figure 3-1: Annotated Version of Figure 2.2 from TELEPERM XS Topical Report

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 NRC-approved SPACE Engineering Tool, which was

approved as part of the TELEPERM XS Topical Report. In the Safety Evaluation Report NRC concluded "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."

The Software Program Manual is predicated on the use of the SPACE Engineering 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.

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 outlines the planning and development activities for all Application Software lifecycle group activities and partially addresses those associated with Operations, Maintenance, and Training, from the requirements phase through the validation phase. The scope of this Software Program Manual relative to NRC BTP 7-14 is shown in Figure 3-3. 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 of 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 Test Phase activities described in Section 11.9.5. 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.

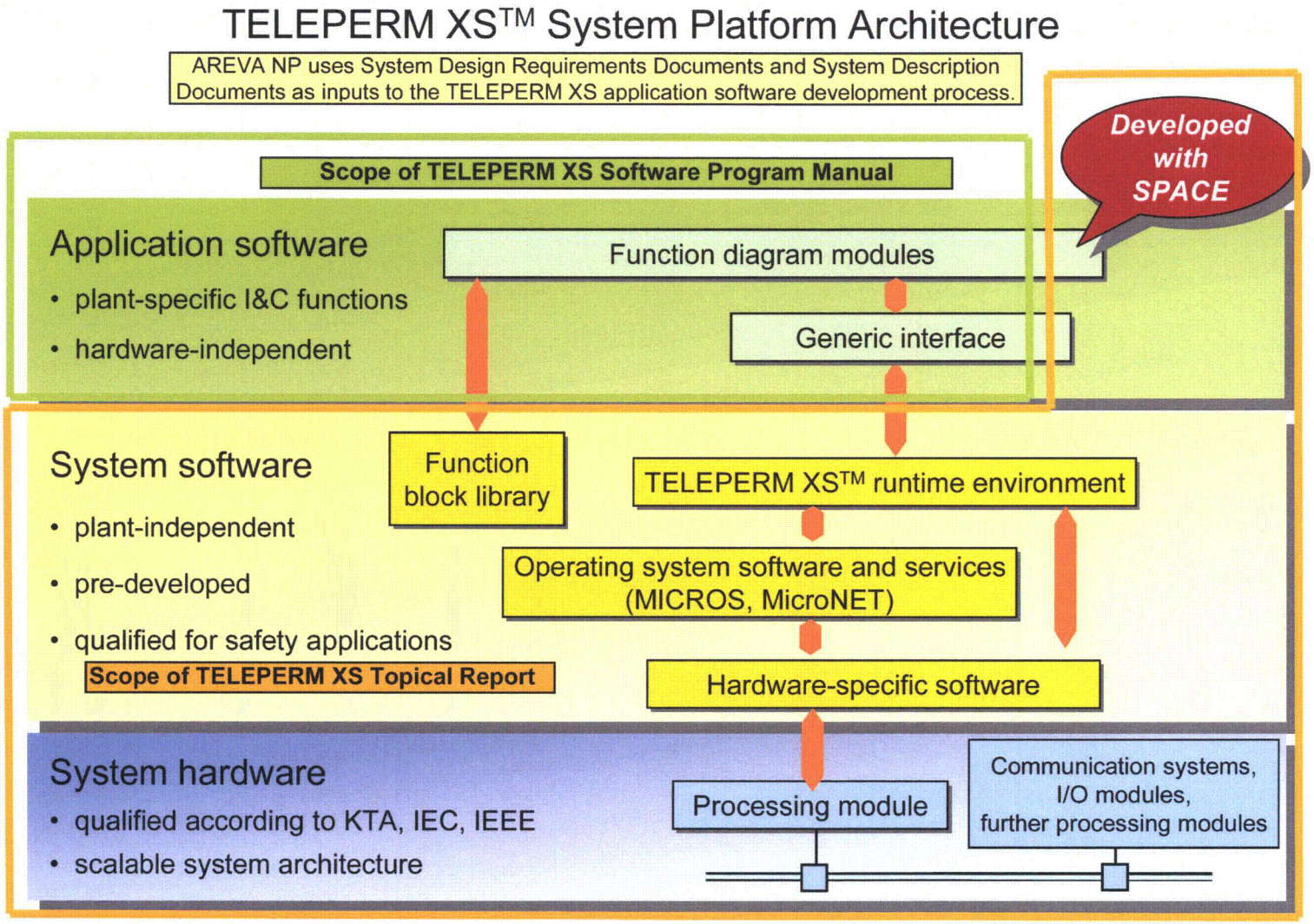


Figure 3-2 - TELEPERM XS Software Program Manual Scope

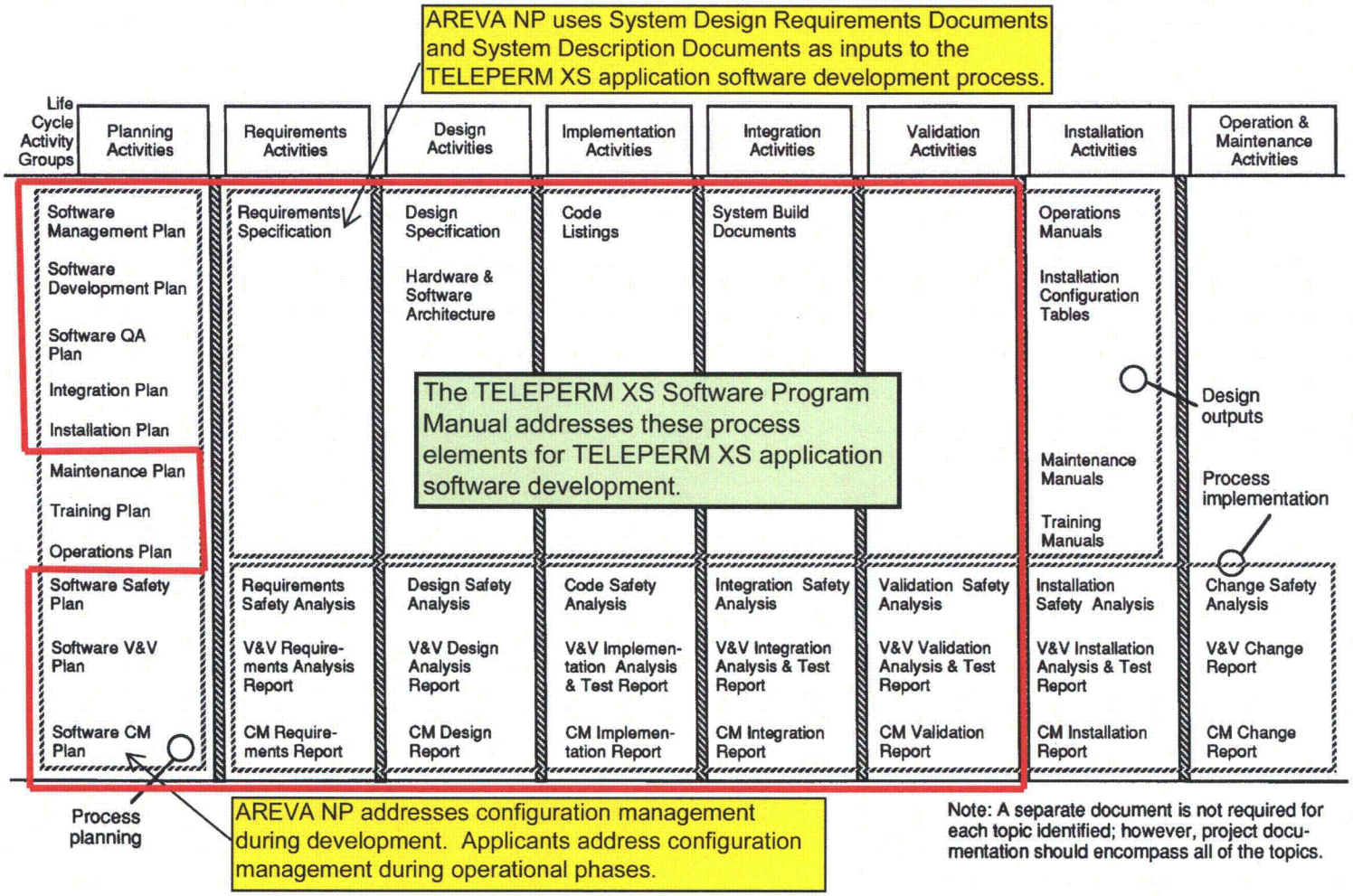


Figure 3-3 - Annotated Version of BTP 7-14 Figure 7-A-1

3.1.5 Relationship to Software and Data 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 security-sensitive information provided to in Appendix C describe the software and data security activities and features associated with the Concept Phase through the Test Phase, as described in Regulatory Guide 1.152 (Reference 7). The following elements of acquisition security controls are addressed for TELEPERM XS technology:

- Elements of AREVA NP supply chain protection for TELEPERM XS Systems are addressed in Appendix C and Section 10.3.2.
- Elements of AREVA NP trustworthiness for TELEPERM XS software development are addressed in Appendix C and Sections 0, 5.0, and 11.0.
- The elements of AREVA NP security testing to ensure that TELEPERM XS software meet all specified security requirements and is free from undocumented software are addressed in Section 13.0 and Appendix C.
- The security features for the TELEPERM XS technology are addressed in Appendix C.
- The elements of AREVA NP controls to prevent tampering of TELEPERM XS software are addressed in Appendix C and Section 12.0.
- The elements of AREVA NP TELEPERM XS System security documentation are addressed in Section 4.5 and Appendix C.
- The elements of TELEPERM XS code analysis are addressed in Appendix C.
- The elements of AREVA NP TELEPERM XS Software configuration management are addressed in Section 12.0.
- The elements of AREVA NP TELEPERM XS Software testing (up through system delivery) are addressed in Appendix C and Section 13.0.
- The elements of AREVA NP verification and documentation of TELEPERM XS security design features are addressed in Appendix C and Section 11.0.

Applicants using the TELEPERM XS technology for safety-related projects will need to address the software and data security items for the Installation, Checkout and Acceptance Testing, Operation, Maintenance, and Retirement Phases. The applicant must address software and data 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 provides the basic organizational structure that governs software quality.

The specific organization for a project is described in the Project Plan for the project. 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.

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 Verification and Validation organization and activities are described in Section 11.0.

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.

A typical project organization chart is shown in Figure 3-4. The organizational titles used are considered representative and may change over time. Regardless of the

organization changes made, the independence of the Verification and Validation group from the design organization and the independent reporting relationships for the QA organization will always be maintained

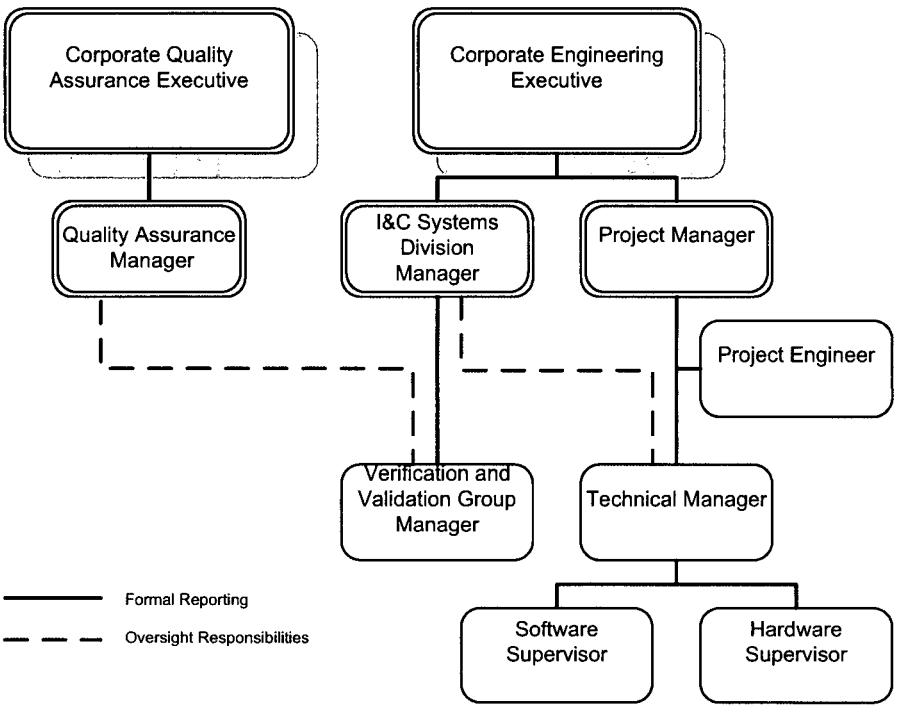


Figure 3-4: Typical Project Organization Chart

3.3 Roles and Responsibilities

The responsibilities for TELEPERM XS projects are described in the following sections.

3.3.1 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 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 program manager executes the project specific training plans correctly.

The Technical Manager is responsible for the execution of the Software Safety Plan, as described in Section 10.2. Various parts of the project organization perform the analyses listed in Section 10.3. A separate software safety organization is not used to implement these activities.

Additional responsibilities of the Technical Manager are described in Sections 5.2, 10.2, 11.3, and 12.2.

3.3.2 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 61). 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 NP 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.

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 3.5.2, 4.5.1, 5.14, 10.2, 11.7, and 12.2.

3.3.3 Project Engineer

The Project Engineer is the AREVA NP lead technical contact for a TELEPERM XS project. The Project Engineer supports the Project Manager responsible for day-to-day and overall management of specific technical areas of the project, including scope, schedule, cost, safety, and quality aspects of the designated technical area. On a small project, the functions of Project Manager and Project Engineer may both be performed

by the Project Engineer; on a large project there may be several Project Engineers supporting the Project Manager.

Additional responsibilities of the Project Engineer are described in Sections 3.5.2, 12.2.2, and 12.4.2.

3.3.4 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.

The Software Supervisor is responsible for the software integration and installation activities for TELEPERM XS projects described in Sections 6.0 and 7.0. A separate software integration organization or software installation organization is not used to implement these activities.

Additional responsibilities of the Software Supervisor are described in Sections 3.5.2, 5.10.1, 10.2, 12.2, and 12.3.5.

3.3.5 Lead Software Designer

The Lead Software Designer for a project reports to the Software Supervisor. The Lead Software Designer leads the Software Engineering group in the preparation and generation of the Software Requirements Specification (SRS), related analyses, Software Design Description (SDD), and other software products.

The Lead Software Designer resolves any anomalies reported by the customer on systems already turned over to the customer.

3.3.6 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 engineers assigned to a project report functionally to the Lead Software Designer and administratively to the Software Supervisor.

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

3.3.7 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 direction 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.

3.3.8 Training Program Manager and Instructors

The training program manager 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.

3.3.9 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. Additional support can provided by the specialized hardware design group of AREVA NP GmbH 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 (Reference 34), as endorsed by Regulatory Guide 1.168 (Reference 9). 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 dotted line relationship to the QA group establishes an independent reporting channel for the Verification and Validation group, which provides an alternate path to escalate problems that are not satisfactorily addressed through the project reporting relationship.

The Verification and Validation group has sufficient authority and resources (budget, staff, etc.) 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 3.5.2, 5.2, 10.2, 11.3, 11.8, and 12.2.

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

Additional responsibilities of the QA organization are described in Sections 5.2, 11.3 and 12.2.2.

3.4 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. The planning activities for a TELEPERM XS Application Software project are documented in the Project Plan that is developed for the specific project.

3.5 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 (Reference 1) 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.

3.5.1 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.

Items from the Open Items list (discussed in Section 3.5.2) are reviewed for conditions adverse to quality and safety, which are entered into the Corrective Action Program. Additionally, problems identified after delivery (see Section 3.5.6) are entered into the Corrective Action Program. Audit findings identified by NUPIC are resolved through the AREVA NP 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 (Reference 66) and affected customers are notified.

The Corrective Action Program is implemented as described in Reference 67.

3.5.2 Open Items Process

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.

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. Changes to project hardware, software, and test design documentation that may be required as a result of an Open Item issue, are processed in accordance with the quality assurance, design, and configuration control procedures.

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. 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. The Verification and Validation engineer independently assesses the design process and the project risk associated with software Open Items at the end of each verification and validation phase of the software development life cycle, as described in Section 11.9.

The Project Engineer and the appropriate design supervisor (i.e., hardware and/or software) approve the release of each Open Item for implementation. Both the Project Managers of the customer and AREVA NP must be involved with the resolution of Open Items that result in the modification of the functional requirements. The Verification and Validation Group verifies and validates the implementation of Open Item solutions if the Open Item is associated with Software or a Hardware/Software issue. The Verification and Validation Group approves the resolution of Open Items that require re-testing.

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 as QA records in the AREVA NP Records Management System at the end of the project as a part of final documentation.

3.5.3 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 63) 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.

3.5.4 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 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.

3.5.5 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 process describes in Section 3.5.2.

3.5.6 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.

3.6 Use of Industry Standards

The industry standards endorsed by NRC, as well as other industry standards used by AREVA NP for the TELEPERM XS Application Software development process, are listed in Table 3-1. The Topical Report sections where the specific use of each standard is described are also listed in the Table.

The use of 'conforms to' in this Topical Report means that all mandatory actions (indicated by "shall") in the industry standards endorsed by NRC are met, with the possible exception of administrative requirements for documents (e.g., placement of

signatures, use of coversheets, and location of reference sections, etc.) that conflict with AREVA NP Records Management System requirements.

For standards not endorsed by NRC, the use is described in this Topical Report. In some cases, AREVA NP has chosen to conform to the standard in the same manner as a standard endorsed by NRC. In other cases, AREVA NP has chosen to use the standard as a guideline; however, full conformance is not claimed and alternative approaches that meet the intent of the standard may be used, as described in this report.

Table 3-1: List of Industry Standards Used for Application Software Development

Standard	NRC Endorsement	Report Sections
IEEE Std 379-2000	Regulatory Guide 1.53, Revision 2	10.3.6
IEEE Std 352-1987	Not endorsed	10.3.6
IEEE Std 577-1976	Not endorsed	10.3.6
IEEE Std 603-1991	Regulatory Guide 1.153, Revision 1	4.4 10.3.6
IEEE Std 730-2002	Not endorsed	5.1 5.4 5.5.1
IEEE Std 828-1990	Regulatory Guide 1.169	5.5.2 5.5.3 12.1.1
IEEE Std 829-1983	Regulatory Guide 1.170	4.5.8 5.6.1 13.6
IEEE Std 830-1993	Regulatory Guide 1.172, Revision 2	4.5.4 5.9.1.1

Standard	NRC Endorsement	Report Sections
IEEE Std 1008-1987	Regulatory Guide 1.171	4.5.8 13.2.2 13.3 13.6
IEEE Std 1012-1998	Regulatory Guide 1.168, Revision 1	3.3.9 4.4 6.0 11.0 13.0
IEEE Std 1028-1997	Regulatory Guide 1.168, Revision 1	5.4 5.5
IEEE Std 1042-1987	Regulatory Guide 1.169	12.1.1 12.2.2
IEEE Std 1063-2001	Not endorsed	4.8
IEEE Std 1074-1995	Regulatory Guide 1.173	4.3 4.5 8.0 Appendix A
IEEE Std 1228-1994	Not endorsed	10.0
IEEE Std 7-4.3.2-2003	Regulatory Guide 1.152, Revision 2	0 5.1 5.9.2 5.14.1 11.2 11.3 11.12 12.1.1

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 approved as an integrated platform. The TELEPERM XS Topical Report describes how the platform software and hardware modules were qualified to work together in a typical system arrangement with representative application software developed with the SPACE Engineering Tool. 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 Engineering Tool.

The TELEPERM XS System Software, Function Block library and the SPACE Engineering Tool Code Generators were developed in accordance with the software lifecycle process described in Section 3 of the TELEPERM XS Topical Report. This generic software was developed using the procedures described in Section 5 of the TELEPERM XS Topical Report.

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 library to generate the Application Software provides a large experience base for the standard modules. The use of the approved automatic code generators (part of the SPACE Engineering Tool) eliminates an important human error source by eliminating conventional software development and code generation.

The SPACE Engineering Tool also eliminates human errors associated with the translation of requirements to code and the introduction of complexity by engineers trying to optimize application coding. Application 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 the SPACE Function Diagrams, 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 and Function Diagram Groups. Function Diagram Group Modules contain code resulting from automatic code generation of Function Diagrams being engineered on the SPACE Engineering Tool. The Function Diagram Group Modules implement the code for the engineered application specific I&C functions. The code consists of a sequence of calls to standard Function Block library functions, as connected for the specific I&C function in the SPACE Function Diagrams. Function Diagram Group Modules also contain code resulting from automatic code generation. The Function Diagram Group code defines call and data interface to all Function Diagram Group Modules running on one TELEPERM XS safety processor. The Function Diagrams in a Function Diagram Group all operate on the same cycle time. A maximum of two Function Diagram Group Modules can be defined for a processor.

The TELEPERM XS platform has been approved 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 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.

4.1.1 TELEPERM XS Platform Changes

TELEPERM XS Systems are assembled using hardware and software modules developed in accordance with the design and qualification methods described in the TELEPERM XS Topical Report. Changes are made to the hardware and software modules for various reasons, including parts obsolescence, operating experience, technology advancements, and new capability needs. Changes to the development

procedures are made to address new requirements or added details to standardize work, operating experience, enhancements in programming guidelines, and alignment with corporate document guidelines.

NRC approved the initial set of TELEPERM XS development procedures and hardware and software modules as part of their approval of the TELEPERM XS Topical Report. NRC may have also approved additional hardware and software modules as part of their approval of license amendment requests for TELEPERM XS projects or through Inspection, Test, and Analysis Acceptance Criteria/Design Acceptance Criteria closure related to new build projects.

AREVA NP can use newer TELEPERM XS hardware and software modules, and associated development procedures provided that the TELEPERM XS Platform Change Control Process described in Appendix D is followed.

4.1.2 Change Evaluation Records

AREVA NP will maintain records of changes to the TELEPERM XS platform or project development procedures used for TELEPERM XS projects in the U.S. in accordance with the AREVA NP Records Management Program. These records must include a written evaluation which provides the bases for the determination that the TELEPERM XS platform or project development procedure change met the criteria listed in Appendix D.

AREVA NP will also provide periodic addenda to the TELEPERM XS Topical Report that updates the list of hardware modules, software modules, and development procedures approved for use in U.S. projects. The addenda will contain a brief summary of the basis for approval (e.g., prior NRC approval or AREVA NP evaluation) and reference to the approval documentation to facilitate any NRC audit of these records.

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 approved automatic code generators in the SPACE Engineering Tool. The development of TELEPERM XS Application Software is predicated on the use of the SPACE Engineering 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 simulation 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 SRS. The SRS becomes the basis for the next detailed level document called the 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. The SPACE Engineering Tool was 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 Engineering Tool qualification is provided in Appendix B.

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

- **cmp_code** can be used for verification of the scope of a modification in Application Software code generated after implementing a specification change in the SPACE project database.
- **cpuload** is used for load analysis for TELEPERM XS processing units.
- **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.
- **fde** is the SPACE Function Diagram editor. It provides a graphical user interface for engineering I&C systems. I&C functions implemented in software are specified on function diagrams. The associated hardware is represented on cabinet diagrams, which provide information on the exact arrangement of the hardware components in the racks and in the I&C cabinet , and network diagrams, which show the communication routes. All data from the I&C project are saved in a database. The functions and topology are engineered with the resources of a formal graphical specification language using a predefined set of blocks (symbols) with defined functionality.
- **fdprint** is used for printing Function Diagrams for the system to support review meetings and documentation requirements.
- **FunBase** is 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. The **FunBase** tool is used to produce the SDD. The output from this tool is reviewed in accordance with the design control process.

- **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.
- **loops** can be used to check whether a unique sequence can be determined for all Function Blocks and Function Diagrams and whether all signals from signal acquisition through to the output modules are connected.
- **l2cpconfig** is a tool used for parameterizing a L2-CP module and for outputting the parameters from the L2-CP module.
- **netload** is used for load analysis for TELEPERM XS local area network connections.
- **rediff** is a 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** is an authentication tool used for registering software configurations in directories. It creates CRC checksums 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 Application Software Releases.

- **scanmic** is an authentication tool used for registering the software configuration in loadable MIC files. 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. 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.
- **smt** is the Service Monitor Test Server, which enables monitoring and maintenance tasks for the online system that are automated using scripts to be tested before the scripts are actually used.
- **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.
- **swparams** may be used for listing software parameters and / or checking the specification data for compliance with the same rules that are also used by the Function Diagram Group code generator for consistency check in the course of code generation.

The SPACE Engineering Tool development and release are controlled under the same QA program, configuration management plan, and software lifecycle development process described in Section 3.2 of the TELEPERM XS Topical Report. NRC reviewed the development procedures and the QA program as part of the review of the TELEPERM XS Topical Report. NRC approved the TELEPERM XS Topical Report in a safety evaluation report issued in May 2000. New tools may be added or tools may be

replaced or modified in the course of maintaining TELEPERM XS SPACE Toolset in accordance with software lifecycle development process.

The handling of the SPACE Engineering Tool by AREVA NP is described in Section 12.3.2 of this Topical Report. The SPACE Engineering Tool is used to support the Application Software design process and conforms with the requirements in IEEE Std 7-4.3.2-2003 clause 5.3.2 (Reference 23), as endorsed by Regulatory Guide 1.152. The output of the Application Software design process developed using these tools is subjected to the independent verification and validation activities described in Sections 11.9 and 11.10 of this Topical Report.

Application Software is developed using the SPACE Engineering 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 Engineering Tool.

The use of the SPACE Engineering Tool ensures that no unused code is inserted in the Application Software. The SPACE Engineering Tool creates software based on the Function Diagrams and Network Diagrams. The qualification of the SPACE Engineering Tool is described in Appendix B. The qualification process established that the SPACE Engineering Tool consistently produces exactly the software specified by the Function Diagrams and Network Diagrams. The deterministic nature of the TELEPERM XS requires that all elements of the Application Software are executed every processing cycle.

Logical 'software integration' occurs at this stage. The project-specific TELEPERM XS system is developed from qualified hardware and software modules using the NRC-approved SPACE Engineering Tool.

Verification of the SPACE Function Diagrams by the Verification and Validation Group is facilitated by the use of a standard symbolic language associated with the object-oriented design tool (SPACE). The I&C functionality can be fully assessed by verification of SPACE Function Diagrams. This check of the Functions Diagrams is equivalent to code verification in other code development systems. The code generation verification checks performed by the SPACE tool can be readily verified.

Physical integration of the Application Software integration with the project-specific TELEPERM XS hardware occurs during the FAT preparation stage, when the project-specific Application Software is loaded on the TELEPERM XS processors. The Application Software Integration Plan is described in Section 6.0. 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 Application Software produced by SPACE Engineering Tool 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 Engineering Tool, the software is verified in various ways throughout the software development lifecycle, as described in Section 11.0. Validation is accomplished with Application Software Integration and System and Acceptance Testing, either using a NRC approved simulation test tool or in the factory

acceptance test field, as described in Section 13.0. This testing is designed to ensure the Application Software generated with the SPACE Engineering Tool 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 Application Software development life cycle is a classic waterfall model. This development model has been chosen based on the years of experience in implementing TELEPERM XS projects in the international market. This model is appropriate for safety-related nuclear power plant projects, since the nuclear safety requirements are generally well known and stable at the time of project implementation.

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

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. The Software Program Manual has been

mapped to the set of life cycle activities provided in Institute of Electrical and Electronic Engineers (IEEE) Std 1074-1995 (Reference 38), as endorsed by Regulatory Guide 1.173 (Reference 14). This mapping 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 39). This mapping was done to demonstrate that no mandatory activities have been omitted from the AREVA NP Application Software Life Cycle Model. The alignment for the AREVA NP activities and the two versions is demonstrated in the mapping.

The IEEE Std 1074-1995 mandatory activity to include suitable comments in the source code (item 5.3.4) is considered not applicable to the TELEPERM XS Application Software developed using the SPACE Toolset (described in Section 0). Since the source code is automatically generated and compiled by the SPACE Tool, the comments associated with code development are recorded in the SDD described in Section 4.5.5 and in the Application Software Code Document described in Section 4.5.6.

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.

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

- Summary Reports for Verification and Validation Activities (i.e., Acquisition Support, Planning, Concept, and Requirements)

- Detailed Design
  - Software Design
  - Automatic Code Generation
  - Application Software Integration Test Planning (using an NRC approved simulation test tool)
  - Application Software Integration Test Execution (using an NRC approved simulation test tool)
  - Application Software Integration Test Reporting (using an NRC approved simulation test tool)
  - Software Safety Analyses
  - Continue Software Requirements Traceability
  - Summary Reports for Verification and Validation Activities (i.e., Design and Implementation)

- System Integration and Testing (including FAT)
  - Integration of Hardware and Software
  - Application Software Integration, System and Acceptance Test Planning
  - Application Software Integration, System and Acceptance Test Execution
  - Application Software Integration, System and Acceptance Test Reporting
  - Continue Software Requirements Traceability
  - Summary Reports for Verification and Validation Activities (i.e., Test)

- Installation and Commissioning (as specified by the customer)
  - Installation and Commissioning Test Planning
  - Installation and Commissioning Test Execution
  - Installation and Commissioning Test Reporting
  - Summary 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.

Table 4-1 provides a general description of the inputs, tasks, processes, outputs, and results associated with each TELEPERM XS project phases. These descriptions are illustrative and typical of the kind of input, tasks, processes, output, and results within a TELEPERM XS Project Phase. The project-specific inputs, tasks, outputs, and results can be modified to align with the nature and complexity of the project, as well as the customer requirements.

Table 4-1: TELEPERM XS Project Input, Tasks, Processes, Output, and Results

Phase: 1	Elaboration of Quotation	
Input/Tasks	**Processes**	**Output/Results**
INPUTS • Customer Specification TASKS • Elaboration of proposal, which addresses all customer requirements	• Projects Manual	OUTPUTS • Proposal to Customer RESULTS • Customer Issues Contract • AREVA NP Accepts Contract
Phase: 2	**Project Start-Up**	
Input/Tasks	**Processes**	**Output/Results**
INPUTS • AREVA NP Contract Acceptance TASKS • Development of project schedule • Development of work breakdown structure • Assign project-manager • Assign project team members/discipline leads • Develop project-specific Software Plans	• Projects Manual • Applicable AREVA NP procedures for preparing and/or processing customer, A/E and supplier documents	OUTPUTS • Project Schedule • Work Breakdown Structure • Project Plan • Project Organization Chart • Project Defining Documents • Project-specific Software Development Plans RESULTS • Project Configuration Controls established • V & V concept established • Conceptual Design Review Board (as required)

Phase: 3	Basic Design	
Input/Tasks	**Processes**	**Output/Results**
INPUTS • Project Start-Up Phase TASKS • Data-Gathering – Interfaces – Plant information • System drawings • Classification of alarms • Function behavior • Existing Documents • Preparing and Releasing Customer Design Input Documents – Contract – Specifications – Functional Requirements – Inputs/Output parameters – Accuracy/Response requirements • Conduct preliminary Design Review (as required)	• Projects Manual • Software Program Manual • Software Configuration Management Plan • Software Verification and Validation Plan • Software Quality Assurance Plan • Software Safety Plan • Implementing Procedures • AREVA NP procedures for preparing and processing documents	OUTPUTS • Documentation and Evaluation of "As-Is" State • Overall I&C Design Concept • Software Requirements Specification • ID Coding Concept • Periodic Test Concept • O&M Concept • Preliminary System Descriptions • System Architecture • Interface Specification • FMEA Baseline • Input to Customer (as needed) • Application Software System and Acceptance Test Plan • V&V Reports RESULTS • Basic Design Freeze • Customer approvals • Conformed Customer Design Input Documents to AREVA NP Basic Design Documents • Preliminary Design Review Board (as required)

Phase: 4	Detailed Design	
Input/Tasks	**Processes**	**Output/Results**
INPUTS • Basic Design with Conformed Customer Design Input Documents to AREVA NP Basic Design Documents TASKS • Detailing of the Concepts of the Basic Design • Combining of the described concepts in one document, which describes all aspects for the new I&C system • Preparing and releasing detailed Design Documents • Conform customer design input documents with AREVA NP detailed design documents • Conduct Final Design Review • V&V activities	• Projects Manual • Software Program Manual • Software Configuration Management Plan • Software Verification and Validation Plan • Software Quality Assurance Plan • Software Safety Plan • Implementing Procedures • AREVA NP procedures for preparing and processing documents	OUTPUTS • System Description • Software Design Description • Application Software Integration Test Plan • Code Documentation • Code Generation • Application Software Integration Test Specifications and Procedures • Application Software System and Acceptance Test Specifications and Procedures • Test Results from Simulation Testing (if performed) • V&V Reports • FMEA • Response time analysis RESULTS • Conformed customer design input documents with AREVA NP detailed design documents • Design Freeze Point • Final Design Review Board • Release for Hardware Manufacture

Phase: 5	Manufacturing	
Input/Tasks	**Processes**	**Output/Results**
INPUTS • Detailed Design Hardware Documents – Hardware Design – Detailed Interface Description – Hardware Design drawings – Procurement documents – Approved Supplier Manufacturing and Test Procedures TASKS • Hardware Manufacturing • Approval of Supplier Manufactured Tested, Hardware, and Required Supplier Hardware Documentation • Design of the cabinets • Design of the internal wiring	• Projects Manual • AREVA NP procedures for preparing and processing documents	OUTPUTS • Cabinets with all necessary modules and hardware • External interface wiring diagrams • Internal Wiring Diagrams • Hardware Quality Assurance Data Package • Hardware Operations and Maintenance Manual RESULTS • Hardware Tests Complete • Tested Hardware Ready for Software Installation • Approved Hardware Documentation

Phase: 6	Testing	
Input/Tasks	**Processes**	**Output/Results**
INPUTS • Hardware Manufacturing Complete • Simulation Testing (if performed with approved simulation tool) • Approved Test documents • Physical Audit TASKS • Equipment Checkout and Prerequisite Testing • Connection of Test Equipment • Loading of TELEPERM XS System and Application Software • Integration Testing (if not performed with approved simulation tool) • Performance of System and Acceptance Tests	• Projects Manual • Software Program Manual • Software Configuration Management Plan • Software Verification and Validation Plan • Software Quality Assurance Plan • Software Safety Plan • Implementing Procedures • AREVA NP procedures for preparing and processing documents	OUTPUTS • Test Summary Test Report • Final V&V Report • Hardware/Software Quality Assurance Data Package • Release to ship System and Documentation RESULTS • Validation of Software Performance • Approved Hardware/Software Documentation • System shipment to site

Phase: 7	Installation and Commissioning (as specified for AREVA NP by customer)	
Input/Tasks	**Processes**	**Output/Results**
INPUTS • Receipt Inspection Plan • Site Acceptance Test Plan, Procedures, and Scripts • Installation Schedule • Installation Procedures • Regulatory approvals • Customer Plant Configuration Management approval • Post Installation Test Plan, Test Procedures, and Test Scripts • Test Equipment TASKS • Receipt Inspection • Site Acceptance Tests Prior to installation • Mechanical and Electrical Installation in the plant • Validation of correct connection wiring inputs, outputs, power supply • Post Installation Test Plan and Procedure, Scripts • Testing of cabinet-specific alarms and their integration into the control boards • Testing of inputs/output signals • Testing of the Gateway functions in combination with Plant Computer • Post Installation Test	• Projects Manual • As specified for AREVA NP by customer	OUTPUTS • TELEPERM XS approved Receipt Inspection Report • Approved Site Acceptance Testing Processes • Approved Site Acceptance Test Report • Approved Installation Processes • Approved Installation Documents • Approved Post Installation Test Documents • Approved Post Installation Test Report • Return to Operation Document • Final Software Release • Software Operations and Maintenance Manual RESULTS • TELEPERM XS System inspected and ready for installation • TELEPERM XS System installed and connected to plant • TELEPERM XS System ready for post installation testing • Fully Functional System (Commissioned) • System turnover for Customer Operation

Phase: 8	Final Documentation	
Input/Tasks	**Processes**	**Output/Results**
INPUTS • Commissioned System • Commissioned System Documents TASKS • AREVA NP Project closeout activities • Customer required project closeout activities • Transmittal of customer required final documentation	• Projects Manual	OUTPUTS • Handover to customer the required final documentation • AREVA NP required closeout records released to Records Management RESULTS • Project completion • Documentation Complete

The project phases used for TELEPERM XS projects described above correlate to the Verification and Validation phase activities described in Section 11.9, as shown in Table 4-2.

Table 4-2: TELEPERM XS Project Phase Correlation

• TELEPERM XS Project Phase	• TELEPERM XS Verification and Validation Phases
• Basic Design	• Concept
	• Requirements
• Detailed Design	• Design
	• Implementation
• Test	• Test
• Installation and Commissioning	• Installation and Checkout

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.

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.

Section 4.5 of this Software Program Manual, describes the required documents.

Section 3.5 of the Software Program Manual, describes procedures to ensure that software errors and failures are promptly acted upon in a uniform manner.

4.4 Software Classification

IEEE Std 1012-1998 provides software integrity scheme; however, the standard does not mandate its use. AREVA NP has chosen to use a different software integrity scheme. The comparison of the IEEE Std 1012-1998 software integrity scheme and the AREVA NP software integrity scheme is shown in Table 4-3.

Table 4-3: Software Integrity Scheme Comparison

Level	IEEE Std 1012-1998 Scheme	AREVA NP Scheme
4	Selected function affects critical performance of the system. (Criticality: High) Annex B Example: An error to a function or system feature that causes catastrophic consequences to the system with reasonable, probable, or occasional likelihood of occurrence of an operating state that contributes to the error; or critical consequences with reasonable or probable likelihood of occurrence of an operating state that contributes to the error.	Software that directly controls safety-related components or systems which are necessary to ensure: • The integrity of the reactor coolant pressure boundary • The capability to safely shut down the reactor and maintain it in a safe shutdown condition • The capability to prevent or mitigate the consequences of an accident that could result in potential offsite exposure in excess of 10 CFR Part 100 guidelines (Reference 5)
3	Selected function affects important system performance. (Criticality: Major) Annex B Example: An error to a function or system feature that causes catastrophic consequences with occasional or infrequent likelihood of occurrence of an operating state that contributes to the error: or critical consequences with probable or occasional likelihood of occurrence of an operating state that contributes to the error; or marginal consequences with reasonable or probable likelihood of	Software which does not meet the definition of Level 4, however: • Is used to perform engineering design or performance calculations for safety-related structures, systems, or components • Is used to perform a function, which is significant to safety or supports a safety-related structure, system, or component

Level	IEEE Std 1012-1998 Scheme	AREVA NP Scheme
	occurrence of an operating state that contributes to the error.	
2	Selected function affects system performance, but workaround strategies can be implemented to compensate for loss of performance. (Criticality: Moderate) Annex B Example: An error to a function or system feature that causes critical consequences with infrequent likelihood of occurrence of an operating state that contributes to the error; or marginal consequences with probable or occasional likelihood of occurrence of an operating state that contributes to the error; or negligible consequences with reasonable or probable likelihood of occurrence of an operating state that contributes to the error.	Software which does not meet the definition of Level 4 or Level 3, however: • Is used to support Technical Specification compliance, Safety Parameter Display System compliance, plant security, station emergency response support, or referred to in the Final Safety Analysis Report • Directly controls or performs engineering designing or performance calculations for non-safety-related structures, systems, or components, or systems essential to plant operation • Is important to comply with regulatory requirements / commitments as required by law and where failure to operate as expected may have indirect effect on nuclear plant safety
1	Selected function has noticeable effect on system performance but only creates inconvenience to the user if the function does not perform in accordance with requirements. (Criticality: Low) Annex B Example: An error to a function or system feature that causes critical consequences with infrequent likelihood of occurrence of an operating state that contribute, to the error: or marginal consequences with occasional; infrequent occurrence of an operating state that contributes to the error: or negligible consequences with probable, occasional, or infrequent likelihood of occurrence of an operating state that contributes to the error.	Software which does not meet the definition of Level 4, Level 3, or Level 2. Level 1 software is important to the continued operation of the plant and whose failure to perform as expected will not affect nuclear safety. Such software may control balance of plant systems, trend or perform analysis of plant operational data, or provide operation or maintenance information in support of decisions regarding plant activities. Level 1 software may also generate output that verifies compliance with regulations unrelated to nuclear safety such as state, local, and environmental laws.

Level	IEEE Std 1012-1998 Scheme	AREVA NP Scheme
0	Not defined.	Software which cannot be categorized as Level 4, Level 3, Level 2, or Level 1. This level is used to show a classification process has been performed.

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 27). All Application 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 as required by the guidance of IEEE Std 1012-1998, as endorsed by Regulatory Guide 1.168.

4.5 Software Development Documentation

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

4.5.1 Project Plans

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 project topics such as controls, budget, quality plan, human resources, communications plan, risk management, special development tools, procurement/contract management, and close out plan. The planning activities for a TELEPERM XS Application Software project are documented in the Project Plan that is developed for the specific project. The project plan identifies the project-manager and the assigned project team leads.

Project schedules are developed and maintained using standard project management tools. The project schedule identifies the key project milestones and is based on a detailed work breakdown structure, including reviews, surveillances, and audits. Technical resources are assigned to the project based on the resource requirements defined in the project schedule.

4.5.2 System Requirements Specification

A SysRS 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 SysRS may be jointly developed by AREVA NP and a 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 software and data security),
- Reliability, maintainability, testability, and fault accommodation, and
- Data export.

For upgrade projects, the SysRS may also derive requirements from a customer specification.

4.5.3 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.

4.5.4 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 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 (Reference 32), as endorsed by Regulatory Guide 1.172 (Reference 13). 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.

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.

4.5.5 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 is comprised of various views of the Application Software, including an overview of the system architecture and a top-level presentation of the important system functions. The SDD describes the important features in the design, such as how the various Function Diagrams interact. Other views include database tables listing the changeable parameters, information signals (with attributes), and communication interfaces. The information provided in this document serves as design input for the detailed Application Software Code.

The SDD logic diagrams can be drawn directly using the SPACE Engineering Tool for small scope, simple projects or using the **FunBase** tool for larger scope, more complex projects. If the **FunBase** tool is used, these diagrams are manually redrawn in the SPACE Engineering Tool, which is then used to automatically generate the code.

4.5.6 Application Software Code Document

The Application Software Code Document is a graphical representation of the TELEPERM XS Application Software developed using the SPACE Engineering Tool. The Application Software Code document provides the complete set of Function Diagrams for the project. The Application Software code is automatically generated from the Function Diagrams with the SPACE Engineering Tool code generators. The Function Diagrams in the Application Software Code Document are exact printouts from the SPACE project database and are in the same format as seen from the SPACE Engineering Tool.

4.5.7 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 identified requirement is traceable backwards to the system requirements or design bases that it satisfies and forward traceable to subsequent design outputs. 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 used to support the requirements traceability analyses performed in each phase of the software lifecycle process, as described in Section 11.9.

The Software Requirements Traceability Matrix is maintained throughout the software development process. The following Software Requirements Traceability Matrix documents will be produced:

- Requirements Traceability Matrix Plan that documents the project specific scope of work as it pertains to tracing.
- Interim Software Requirements Traceability Matrix report that is comprised of all released tracing reports. This interim report will be produced at the completion of the Detailed Design Phase.
- Final Software Requirements Traceability Matrix report that is comprised of all released tracing reports. This final report will be produced at the completion of the System Integration and Testing Phase.

4.5.8 Software Test Plans

Application Software testing follows written test plans that conform to the guidance of IEEE Std 829-1983 (Reference 31), as endorsed by Regulatory Guide 1.170 (Reference 11). The Applications Software Integration and System and Acceptance Test Plans prescribe the scope and approach of the testing activities. They identify 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 plans address the objectives of test coverage.

The Applications Software System and Acceptance Test Plan is prepared during the basic design phase of the software development life cycle, as described in Section 11.9.2. The Application Software Integration Test Plan is prepared during the detailed design phase of the software development life cycle, as described in Section 11.9.3. These Test Plans are reviewed for completeness, accuracy, and consistency, prior to the commencement of formal testing.

Application Software validation testing is planned and executed in accordance with procedures that conform to the applicable guidance of IEEE Std 1008-1987 (Reference 33), as endorsed by Regulatory Guide 1.171 (Reference 12). See Section 13.0 for a description of the complete approach to Application Software testing.

4.5.9 Software Test Documentation

The documentation used to support Application 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
- 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. Preparation of the Software Test Documentation is described in Section 11.9.

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 software and data security items associated with the Concept Phase through the Test Phase, as described in Regulatory Guide 1.152.

4.6.1 Software and Data Security Design Features

The TELEPERM XS system meets the nuclear safety requirements for interference-free communication and safety to non-safety system isolations. These TELEPERM XS design features, which provide control of access, communication independence, safety to non-safety system isolation, and interference-free communication, also address the

software and data security attributes of confidentiality, integrity, and availability. The key software and data 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. As such, some elements of the generic system qualification are not explicitly addressed as software and data security activities in the TELEPERM XS Topical Report and the associated NRC safety evaluation report. The TELEPERM XS software development process controls related to software and data 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 or unidirectional control), which are determined for each project in coordination with customer software and data security controls. Additional information on the TELEPERM XS design features that provide software and data security protection is provided in Appendix C.

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

4.6.2 Software and Data 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 is provided in Appendix C.

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 will support the customer's efforts on the development of security procedures and assessments. Additional requirements for AREVA NP 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 software and data security items for the Installation, Checkout and Acceptance Testing, Operation, Maintenance, and Retirement Phases.

4.7 Documentation Standards

All design documents produced for a TELEPERM XS project are prepared in accordance with the requirements of the AREVA NP Quality Management Manual. The documents also meet the requirements of industry standards where conformance is claimed, with the possible exception of administrative requirements for documents (e.g., placement of signatures, use of coversheets, and location of reference sections, etc.) that conflict with AREVA NP Records Management System requirements. All 10 CFR Part 50 Appendix B documents delivered to the customer are handled in accordance with AREVA NP QA document requirements (Reference 64).

4.7.1 Naming Conventions

The identification (ID) coding standards for the SRS are based on the description of functions in the SysRS. The ID coding of field equipment is based on the ID assigned

for the field devices. ID codes for signals, transmitters, and actuators are described in interface lists and the project-specific ID coding concept.

The Application Software generated by the SPACE Engineering Tool is assigned to a central processor unit ID, which has to be defined in the Function Diagrams. A typical ID scheme is would include identification of unit, redundancy, cabinet, and component.

The SPACE Engineering Tool documents the hardware and software in the form of diagrams, which are identified by ID codes. SPACE diagrams are distinguished by diagram type. The Application Software designed for TELEPERM XS computers is based on the SPACE Function Diagrams. These Function Diagrams are named using an unambiguous ID code. The Function Diagrams are identified by this ID code. Signal connections between Function Diagrams carry the ID code of the source Function Diagram plus a signal identifier.

The SPACE Engineering Tool requires ID codes for signals being exchanged between Function Diagrams (called external signals). These external signals carry the ID code of the Function Diagram they are originating from, plus a signal identifier.

The TELEPERM XS System Software modules are identified by file name, version and date, file size, and CRC checksum.

Application Software developed for a project is identified by a three-digit version number. The first digit of the version number is incremented when the Application Software is migrated to a newer version of System Software. The second digit of the version number is incremented when the proposed changes make significant modifications to the functionality of the software or adds new functions to the software. The third digit of the version number is incremented by one when the proposed changes make minor modifications to the software (i.e., bug fixes, input/output format changes, etc.).

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 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.

The source code for the safety function is created by the code generators in the SPACE Engineering Tool. The SPACE Engineering Tool generates one source code file for each specified Function Diagram. Each source code file contains information about the respective Function Diagram (e.g., date and time of creation). For TELEPERM XS Graphic Service Monitor (GSM) and Qualified Display System (QDS) Application Software, the code is generated using a combination of manually created scripts and Dialogue Screens created using the Qt Designer tool utilizing special GSM and QDS libraries.

4.7.3 Logic Structure Standards

The **FunBase** tool provides a structured representation of the safety logic created to facilitate the design of the Application Software. The **FunBase** tool supports the translation of the software design requirements specified in the SRS into a description of the software structure, software components, interfaces, and data necessary for the implementation of the design into the Application Software.

A typical Application Software structure is shown in Figure 4-1.

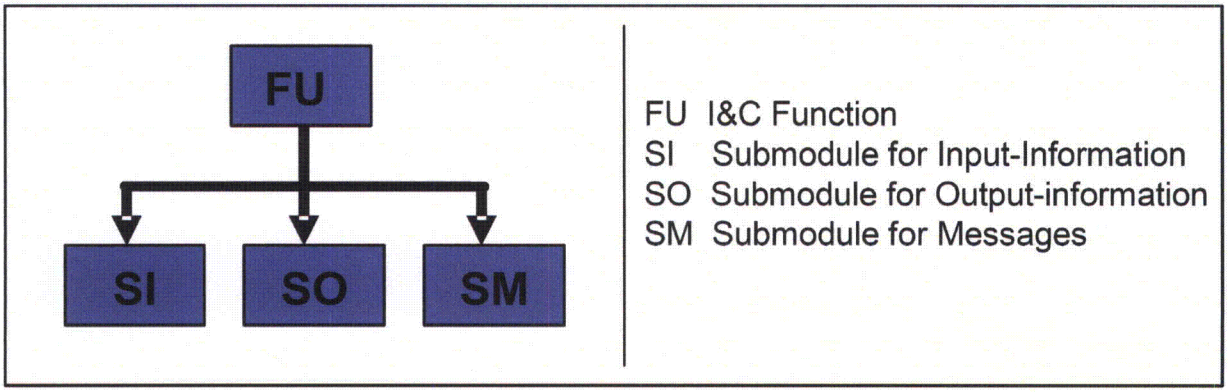


Figure 4-1: Hierarchical Structure of the Application Software Design

Functions (FUs) are designed in accordance with the definition in the SysRS. The FUs are comprised of complete I&C functions in process engineering terms; starting from measured data acquisition (of the initiation or control criteria), information processing (generation of initiation or control signals, logic processing for the generation of actuation or control signals) to the control element actuation logic.

Submodules fulfill the purpose of documenting detailed information separately (i.e., for measured data acquisition or control element actuation logic) in order to achieve a clear overview of the overall function on the FU level. Submodules can be used in different FUs such as when different FUs require the same measuring information or when they actuate the same field components. Submodules are only defined once even when they are used by multiple FUs. Signal connections between submodules and FUs are clearly indicated by submodule inputs and outputs. The following types of submodules can be defined:

- Input Submodules (SI) - SIs are used for the acquisition of process information. In general, they describe the processing of measured data (i.e., conversion to

physical units and the Signal Online Validation of input signals). Input signals, which are used in different FUs, are always described using SIs.

- Output Submodules (SO) - SOs are related to specific actuators or a group of actuators. They process signals from various FUs. They can also acquire external input signals (i.e., checkback signals from field components). SOs perform the prioritization of the control commands coming from the different FUs.
- Monitoring Processing Submodules (SM) - SMs are used to describe the interface to the control room or the TELEPERM XS Gateway. SMs can also acquire external input signals (e.g., Parameter Change Enable Keyswitch).

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 11.9.3.

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 Engineering 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 project 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 Software. The source files are generated in the programming language

ANSI-C. The created source code files are not modified manually; they are subject to the rules described in the Software Configuration Management Plan.

4.8 User Manuals

AREVA NP prepares user and maintenance manuals following the recommendations of IEEE Std 1063-2001 (Reference 37). 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.

5.0 SOFTWARE QUALITY ASSURANCE PLAN

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.

5.1 Purpose

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 29). The 2002 version is an update to the 1998 version, which incorporated additional requirements to be "consistent with Clause 6.20 of IEEE/EIA 12207.1-1997 and obviates the need for Annex A of the 1998 version of the standard." The Software Quality Assurance Plan also conforms to the requirements in IEEE Std 7-4.3.2-2003 clause 5.3.1, as endorsed by Regulatory Guide 1.152.

The audit activities described in Section 5.5 are performed by the independent AREVA NP QA organization. The governing document for these activities 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
- Software Safety Plans
- Design Documents
- Application Code

5.2 Management

The Technical Manager maintains the Software Quality Assurance Plan. The Software Quality Assurance Plan is reviewed by the Verification and Validation group manager and approved by the QA manager and the I&C Systems department manager.

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 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 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 been followed.

The QA group performs the audit activities described in Section 5.5 and verifies that the implementation of QA requirements is in accordance with the Quality Management Manual.

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 into consideration the quality factors listed in BTP 7-14 Section B.3.3, Acceptance Criteria for Design Outputs.

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. See Section 4.5 for a description of Software Development Documentation, Section 11.13 for a description of Verification and Validation Reports, and Section 13.6 for a description of Validation Test Documents.

5.4 Software Reviews

Software reviews are conducted in accordance with IEEE Std 730-2002 and IEEE Std 1028-1997 (Reference 35), 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 detailed in the individual project schedules. A minimum set of reviews are conducted as described in Sections 5.4.1 through 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 may be performed along with the Preliminary Design Review. The Software Requirements Review shall verify that the SRS was created in accordance with the standards listed in Section 4.5.4 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. This activity shall be accomplished by a Technical Review, as defined in IEEE Std 1028-1997. The Technical Manager is the decision maker and determines if the Software Requirements Review objectives have been met. The Verification and Validation group actively participates in the review and evaluation of the SRS. QA monitors and reports on the progress and status of any outstanding corrective actions or Open Items that result from the review.

5.4.2 Preliminary Design Review

The Preliminary Design Review takes place at the end of the Basic Design Phase. 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. This activity shall be accomplished by a Technical Review, as defined in IEEE Std 1028-1997. (For a small project, this review may not be required. This exception shall be taken in the project-specific plan.) The Technical Manager is the decision maker and determines if the Preliminary Design Review objectives have been met. The Verification and Validation group actively participates in the review and evaluation of the system and software architecture documents. QA monitors and reports on the progress and status of any outstanding corrective actions or Open Items that result from the review.

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 SysRS. 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 activity shall be accomplished by a Technical Review, as defined in IEEE Std 1028-1997. The Technical Manager is the decision maker and determines if the Detailed Design Review objectives have been met. The Verification

and Validation group actively participates in the review and evaluation of the SDD and the SPACE Function Diagrams. QA monitors and reports on the progress and status of any outstanding corrective actions or Open Items that result from the review.

5.4.4 Software Verification and Validation Plan Review

A Software Verification and Validation Plan Review shall be performed after the initial Software Verification and Validation Plan is produced. The Software Verification and Validation Plan Review shall evaluate the adequacy and completeness of the verification and validation methods defined in the Software Verification and Validation Plan. This activity shall be accomplished by a Management Review, as defined in IEEE Std 1028-1997. The Software Verification and Validation Plan Review is carried out by, or on behalf of, the management personnel having direct responsibility for the system. The Verification and Validation group manager is the decision maker and determines if the Software Verification and Validation Plan Review objectives have been met. The Design and Verification and Validation groups provide the information necessary for the management staff to fulfill its responsibilities. QA monitors and reports on the progress and status of any outstanding corrective actions or Open Items that result from the review.

5.4.5 Managerial Reviews

Managerial Reviews shall be held and are scheduled 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. These activities shall be accomplished by a Management Review, as defined in IEEE Std 1028-1997. These Managerial Reviews are carried out by, or on behalf of, the management personnel having direct responsibility for the system. The management sponsor will designate a decision maker responsible for determining if the Managerial Review objectives have been met. The Design and Verification and Validation groups provide the information necessary for the management staff to fulfill its responsibilities. QA monitors and reports on the progress and status of any outstanding corrective actions or Open Items that result from the review.

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. This activity shall be accomplished by a Management Review, as defined in IEEE Std 1028-1997. The Software Configuration Management Plan Review is carried out by, or on behalf of, the management personnel having direct responsibility for the system. The Technical Manager is the decision maker and determines if the Software Configuration Management Plan Review objectives have been met. The Design and Verification and Validation groups provide the information necessary for the management staff to fulfill its responsibilities. QA monitors and reports on the progress and status of any outstanding corrective actions or Open Items that result from the review.

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. This activity shall be accomplished by a Management Review, as defined in IEEE Std 1028-1997. The Post-Implementation Review is carried out by, or on behalf of, the management personnel having direct responsibility for the system. The management sponsor will designate a decision maker responsible for determining if the Post-Implementation Review objectives have been met. The Design and Verification and Validation groups provide the information necessary for the management staff to fulfill its responsibilities. QA monitors and reports on the progress and status of any outstanding corrective actions or Open Items that result from the review.

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. The QA organization is responsible for ensuring that the lead auditor is provided the appropriate levels of expertise and knowledge sufficient to comprehend the software product being audited and may include technical resources such as Verification and Validation personnel or third party experts, as necessary.

The Software Audits conform to the guidance of IEEE Std 1028-1997, as endorsed by Regulatory Guide 1.168, except where the guidance conflicts with the administrative requirements in the AREVA NP Quality Management Manual, the QA organization implementing procedures, or the Records Management Program Manual requirements (e.g., audit meeting and reporting protocols, placement of signatures, use of coversheets, and location of reference sections, etc.).

5.5.1 In-Process Audits

The reviews, inspections, and requirements tracing activities described in Section 11.9 are credited for conforming with 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 SysRS, SRS, SDD, test plans, specifications, cases, procedures, and results. Also included are code reviews for software not generated by SPACE Engineering Tool. 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 3.5.

This Audit, along with the Functional Audit in Section 5.5.3, serves as a Configuration Audit per IEEE Std 828-1990. This audit shall be performed by the QA organization as an Audit Review, as defined in IEEE Std 1028-1997.

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.

This Audit, along with the Physical Audit in Section 5.5.2, serves as a Configuration Audit per IEEE Std 828-1990. This audit shall be performed by the QA organization as an Audit Review, as defined in IEEE Std 1028-1997.

5.5.4 Software Process Audits

Software Process Audits are conducted annually and are held to verify compliance with applicable software development requirements. The Software Process Audits will be used to review the procedure change evaluation records from Section 4.1.2 to ensure that AREVA NP Operating Instructions implement the requirements of the Software Program Manual. The audits are performed in accordance with the AREVA NP Quality Management Manual.

5.6 Software Testing

The Verification and Validation group is responsible for preparation of the Application Software Integration and System test documents and performance of the validation testing, as described in Section 11.8.

The Application Software Integration Test Plan is prepared to establish that the project-specific Application Software Function Blocks and Function Diagrams have been properly integrated, based on the project-specific system architecture and SDD. The comprehensiveness of the testing effort should ensure that all functionality defined in

the SDD is tested. Application Software Integration Testing is described in Section 13.3.

The Application Software System and Acceptance Test scope includes the following:

- System characteristics (e.g., response times),
- System behavior in case of failures of the components and system parts,
- Correct implementation of the system level I&C requirements and verification of the system interfaces, and
- Test of the specified function of the overall system under selected, simulated operation and failure conditions

The Application Software System and Acceptance Test Plan is prepared to ensure that the functional requirements of the SysRS and SRS are validated. The comprehensiveness of the testing effort should ensure that all testable requirements of the SysRS and SRS are tested. The Application Software System and Acceptance Testing is described in Section 13.5. The FAT enables the customer to determine whether or not to accept the system.

5.6.1 Software 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, as endorsed by Regulatory Guide 1.170.

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 Software 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.

5.6.3 Software 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 3.5.4.

5.7 Standards

Section 15.0 lists the regulations, regulatory guidance, and industry standards that are used in the creation and administration of the TELEPERM XS Software Program Manual. These standards address the topics of design standards, documentation standards, test standards, and software QA products.

The industry standards endorsed by NRC, as well as other industry standards used by AREVA NP for the TELEPERM XS Application Software development process, are discussed in Section 3.6. The section of this Topical Report where the specific use of

each standard is described is identified in Table 3-1. Additional information on documentation standards is provided in Section 4.7. As noted in Section 4.3, the commentary standards for TELEPERM XS code development are described in Sections 4.5.5 and 4.5.6. Software development process metrics are described in Section 11.12.

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 3.5.2 contains further discussion of the details of the Open Items system.

The AREVA NP formal problem reporting and Corrective Action Program 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.

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, as endorsed by Regulatory Guide 1.172. 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 SysRS 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 Tool contains the tools for converting Function Diagrams into software code and includes the source code-generators, such as Function Diagram Group Module and Runtime Environment, and the software for compiling, linking and locating, such as make command. These tools are part of the qualified TELEPERM XS software package.

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 (SysRS, SysDD, SRS, and SDD, including the SPACE diagrams (used to generate the code).
- The TELEPERM XS software tools package (such as SPACE Engineering Tool, 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. This metric approach is used to conform to the requirements in IEEE Std 7-4.3.2-2003 clause 5.3.1.1, as endorsed by Regulatory Guide 1.152.

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.

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.

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 Application Software Integration Tests
- Test specifications, procedures, and test-results of the System and Acceptance Tests (including FAT)
- Software Requirements Traceability Matrix

5.13 Training

Design and Verification and Validation personnel are trained on the provisions of this Software Program Manual, as specified in training requirements prepared in accordance with AREVA NP procedures (Reference 65).

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.

5.14.1 Project 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 62) 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 (including development of appropriate contingencies). 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 requirements in IEEE Std 7-4.3.2-2003 clause 5.3.6, 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 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 Technical 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 Engineering 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 approved 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. Application Software Integration and System and Acceptance 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 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.

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 integration process for a TELEPERM XS consists of three stages: TELEPERM XS Platform integration, project-specific Application Software integration, and project-specific hardware and software integration.

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.

6.1 TELEPERM XS Platform Integration

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. Integration of the generic TELEPERM XS platform is the responsibility of AREVA NP GmbH.

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 purpose of the generic system qualification tests is to validate that the TELEPERM XS Hardware, System Software, and representative Application Software operate correctly in an integrated environment. The system test documentation is listed in Section 8.1.1 of the TELEPERM XS Topical Report. There are no project-specific requirements that

affect the generic approach of integrating TELEPERM XS hardware and software. As such, the project-specific integration efforts are built on this foundation.

6.2 TELEPERM XS Application Software Integration

Application Software is developed using the SPACE Engineering 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 devices 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 automatically generated from the Function Diagrams and Network Diagrams by the SPACE Engineering Tool. Logical 'software integration' occurs at this stage. The project-specific TELEPERM XS system is developed from qualified hardware and software modules using the approved development tools. Integration of the Application Software with the SPACE Engineering Tool is the responsibility of the Software Design Group. The SPACE Engineering Tool is described in Section 0.

Application Software Integration Testing is performed in accordance with the Software Verification and Validation Plan, as described in Section 11.0 to conform to IEEE Std 1012-1998 validation requirements. Application Software Integration Testing is the responsibility of the Verification and Validation Group. The purpose of the testing is to establish that the project-specific Application Software Function Blocks and Function Diagrams have been properly integrated, based on the project-specific system architecture and SDD. The Application Software Integration Testing can be performed in a simulation environment with an approved simulation test tool or it can be performed in the test field as part of the FAT.

Successful completion of the Application Software Integration Testing is the means of determining the success or failure of the Application Software integration effort. Test

discrepancies are reported as described in Section 3.5.4. Test discrepancy metrics are evaluated as described in Section 11.12.

This Application Software Integration Testing is performed either in the Verification and Validation Implementation Phase (as described in Section 11.9.4) or Test Phase (as described in Section 11.9.5). See Section 13.3 for a description of Application Software Integration Testing.

6.3 TELEPERM XS Project Hardware and Software Integration

Physical integration of the Application Software with the project-specific TELEPERM XS hardware occurs during the FAT preparation 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 is responsible for the project-specific hardware and software integration, which is implemented with the Software Generation and Download Procedure.

Project-specific Hardware and Software Integration Testing is performed in accordance with the Software Test Plan, as described in Section 13.0. Hardware and Software Integration Testing is the responsibility of the Verification and Validation Group. The purpose of the testing is to establish that the project-specific TELEPERM XS System

has been configured correctly, is operating properly, and that the prerequisites for validation testing to conform with IEEE Std 1012-1998 requirements have been met.

Successful completion of the Hardware and Software Integration Testing is the means of determining the success or failure of the project-specific hardware and software integration effort. Test discrepancies are reported as described in Section 3.5.4. Test discrepancy metrics are evaluated as described in Section 11.12.

The Hardware and Software Integration Testing is performed in the Verification and Validation Test Phase, as described in Section 11.9.5. See Section 13.4 for a description of Hardware and Software Integration Testing.

7.0 SOFTWARE INSTALLATION PLAN

The Software Installation Plan describes the installation process for TELEPERM XS projects to support FAT. As described in Section 3.1.4, the installation of the TELEPERM XS System in the plant environment is not within the scope of the Software Program Manual. The Software Installation Plan for the plant environment is addressed as a project-specific item.

7.1 TELEPERM XS Software Installation

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.
- Preparation of the Configuration File for the TELEPERM XS communication processors.
- Loading of the TELEPERM XS Application Software onto the Service Unit,
- Loading of the project specific Service Monitor client software onto the Service Unit,
- Loading of the Software onto the TELEPERM XS safety function processor modules in the following order:
  - Loading of the System Segment Software onto the TELEPERM XS processing modules and
  - Loading of the Application Software onto the TELEPERM XS processing modules.
- Loading of the Software onto the TELEPERM XS communication processor modules in the following order:

  - Loading of the System Segment Software onto the TELEPERM XS processing modules and
  - Loading of the L2-CP Firmware onto the TELEPERM XS communication processors used for Profibus communication and parameterization of the L2-CP Firmware or
  - Loading of the H1-CP Firmware onto the TELEPERM XS communication processor used for Ethernet communication.
- Loading of the Application Software onto the TELEPERM XS Gateway.

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,
- 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.

The Software Generation and Download Procedure also includes instructions for producing the documentation for assembling and identifying the TELEPERM XS Application Software Code release.

Software may be installed at several points during the course of the project. The first time an Application Software Release is installed is in preparation for the FAT. The initial software load is made using the TELEPERM XS Maintenance Laptop, since bootstrap loading of any TELEPERM XS processor is not possible via the TELEPERM XS Service Unit because access from Service Unit is not possible without TELEPERM

XS System Software, Application Software, and pre-defined communication links installed. The Maintenance Laptop is also used to configure the L2-CP communication modules. Once the initial software load is complete, the Maintenance Laptop must also be used to install software on a new processor board (e.g., after maintenance replacement) or to install System Software upgrades.

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 Verification and Validation personnel can also install the software during the system test phase. AREVA NP can also install the software prior to installation and commissioning, if requested by the customer.

The Application Software download can be performed as central download from the Service Unit or local download with the Maintenance Laptop. 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).

Command file scripts can be used to facilitate the software installation process. These scripts are listed in the project-specific TELEPERM XS Application Software Generation and Download Procedure. No special controls are required for the software installation scripts because the proper installation is fully verified, as described in Section 7.2.

7.2 TELEPERM XS Software Installation Verification

The SPACE Engineering Tool is used to generate the CRC checksums separately in advance of the download (using the **scanmic** tool). 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 EEPROM of the TELEPERM XS processor. After downloading, the two CRC checksums are compared to verify that the correct version of the 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 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 Application Software CRC checksums stored in the FEPROM of the respective TELEPERM XS processor can be extracted via manual request from the Service Unit and compared against the CRC checksums in Application Software files stored on Service Unit.

The same installation verification methods, performed from the Service Unit, are used for software loaded from Maintenance Laptop or Service Unit.

7.3 TELEPERM XS Service Unit Software Installation

The following software is installed on the TELEPERM XS Service Unit:

- Linux Operating System.

- SPACE Engineering Tool used to generate the Application Software. This software package also includes all precompiled System Software.
- Oracle database used to store the information from the SPACE Engineering Tool.
- TELEPERM XS Gateway Software for generating TELEPERM XS Gateways
- TELEPERM XS Support for Linux, which modifies Linux operating system files to ensure compatibility of the select Linux Operating System and the selected TELEPERM XS SPACE version.

7.4 TELEPERM XS Maintenance Laptop Software Installation

The following software is installed on the TELEPERM XS Maintenance Laptop Computer:

- Linux Operating System (may be different version than Service Unit because of the need to match the Linux version to the hardware of the laptop).
- SPACE Engineering Tool (same version as Service Unit).
- Oracle database (may be different version than Service Unit because of the need to match the Linux version to the hardware of the laptop).
- TELEPERM XS Support for Linux (may be different version than Service Unit because of the need to match the Linux version to the hardware of the laptop).

8.0 SOFTWARE MAINTENANCE AND OPERATIONS PLAN

The Software Operations and Maintenance Plan describes the post-customer delivery software practices. As described in Section 3.1.4, the Software Operations Plan and Software Maintenance Plan are not within the scope of the Software Program Manual. The Software Operations Plan and Software Maintenance Plan are addressed as project-specific action items.

The AREVA NP 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 1.173, for vendor support activities. IEEE Std 1074-1997 Annex A was used as in input for defining the activities to be included in the Software Operations and Maintenance Plan, since there is not any other specific guidance for the plan. The AREVA NP Software Operations and Maintenance Plan is consistent with the portions of the software operations and maintenance planning guidance of BTP 7-14 related to vendor problem reporting. After the system has been turned over to the customer, software modifications and maintenance by AREVA NP are performed in accordance with this plan.

8.1 Purpose

The plan describes the activities and resources that enable AREVA NP to support and maintain TELEPERM XS 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).

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.

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 evaluation will be performed in accordance with the AREVA NP procedure for evaluation and reporting of safety significant issues. 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.

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.

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. As described in Section 3.1.4, implementation of the Software Training Plan is not within the scope of the Software Program Manual. The implementation of the Software Training Plan is addressed as a project-specific action item.

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.

- 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 three-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
- 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. 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 functionally reports to the project organization to ensure that the project-specific customer training meets the customer needs. 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.

9.3 Responsibilities

The AREVA NP customer Software Training Plan describes the training organization, which is directed by the training program manager. The training program manager 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 program manager provides training and directs SMEs in the presentation of training when necessary. Additionally, the training program manager provides the main interface between training, project management, engineering, and the customer to ensure that the necessary transfer of knowledge and skills is provided.

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

10.0 SOFTWARE SAFETY PLAN

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

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 Verification and Validation Group performs the activities listed in Sections 10.3.5, 10.3.9, and 10.3.10. The Verification and Validation Group also reviews the design work associated with the activities listed in Sections 10.3.3, 10.3.4, and 10.3.8.

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.
- 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 project Software Supervisor.
- 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.

The Software Design Group performs the activities listed in Sections 10.3.3, 10.3.4, 10.3.7, and 10.3.8. The Hardware Design Group performs the activities listed in Section 10.3.6 and supports the activities listed in Section 10.3.7.

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 40). 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 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 Engineering Tool. As such, the Software Design group does not write code in the traditional manner; 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 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 approved software development tools and structured engineering analyses. The use of a qualified Function Block library provides a large experience 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 Engineering Tool eliminates both errors of translation and the introduction of complexity by engineers trying to optimize application coding. The use of the SPACE Engineering 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 the NRC-approved SPACE Engineering Tool.

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 conform to 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 meets the intent of the requirement for a preliminary hazards analysis.

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 18). 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 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.

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

10.3.5 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 describing the verification work performed 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.

10.3.6 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 24), as endorsed by Regulatory Guide 1.53 (Reference 6). 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, as endorsed by Regulatory Guide 1.153 (Reference 8). The guidance in Section 4.1 of IEEE Std 352-1987 (Reference 25) 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 26) 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.

10.3.7 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 SysRS or the response time assumed in the design basis accident analyses described in the Design Control 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.

10.3.8 Automatic Code Generation

The Automatic Code Generation is performed at the end of the Detailed Design Phase. The Application Software is programmed using a graphical coding method (i.e., the SPACE Functional Diagram Editor). The use of the SPACE Engineering Tool to produce Application Software improves the reliability of the software. For example, the

use of a standard Function Block Library provides a large experience base for the standard modules. The use of the SPACE Code Generators eliminates an important human error source by eliminating conventional software development and code generation (i.e., errors of translation and the introduction of complexity by engineers trying to optimize application coding). The graphical representation of the TELEPERM XS Application Software in the Application Software Code Document, produced by the SPACE Engineering Tool, provides the complete set of Function Diagrams for the project in a format that readily supports review and verification activities. The result of this graphical coding is then run through the NRC-approved automatic code generators. As described in Section 0, the SPACE Engineering Tool also contains a number of specific tools that are used to identify errors in Application Software. The use of the SPACE Engineering Tool reduces the likelihood of a manual coding error that may pose a risk that is adverse to safety.

Section 13.8 describes the test tools that are used to reduce errors during the performance of testing and support identification of errors in Application Software.

10.3.9 Application Software Integration Testing

The Application Software Integration Test Plan is prepared to ensure that the functional requirements of the software design detailed in the SDD are properly implemented in the SPACE project database. The comprehensiveness of the testing effort should ensure that all functionality defined in the SDD is tested. The Application Software Integration Test Report summarizes the findings of the integration testing. The Software Verification and Validation Plan is further described in Section 11.0. The Software Test Plan is further described on Section 13.0.

10.3.10 Application Software System and Acceptance Testing

The Application Software System and Acceptance Test Plan is prepared to ensure that the functional requirements of the SysRS and SRS are validated. The comprehensiveness of the testing effort should ensure that all testable requirements of the SysRS and SRS are tested. Application Software System and Acceptance Testing

is performed to validate that the functionality of the system meets the design and customer requirements in the fully integrated system. The Application Software System and Acceptance Test Report summarizes the findings of the system and acceptance testing. The Software Verification and Validation Plan is further described in Section 11.0. The Software Test Plan is further described on Section 13.0.

10.4 Documenting and Correcting Safety Hazards

A software safety analysis summary document is prepared to summarize the software safety analysis activities for each TELEPERM XS project. The individual safety analysis activities and the summary are documented as a quality records in the AREVA NP Records Management System.

Any safety hazard discovered in the analyses discussed above is documented using the AREVA NP Corrective Action Program and corrected as described in Section 3.5.1.

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.

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.

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. 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.

11.2 Overview

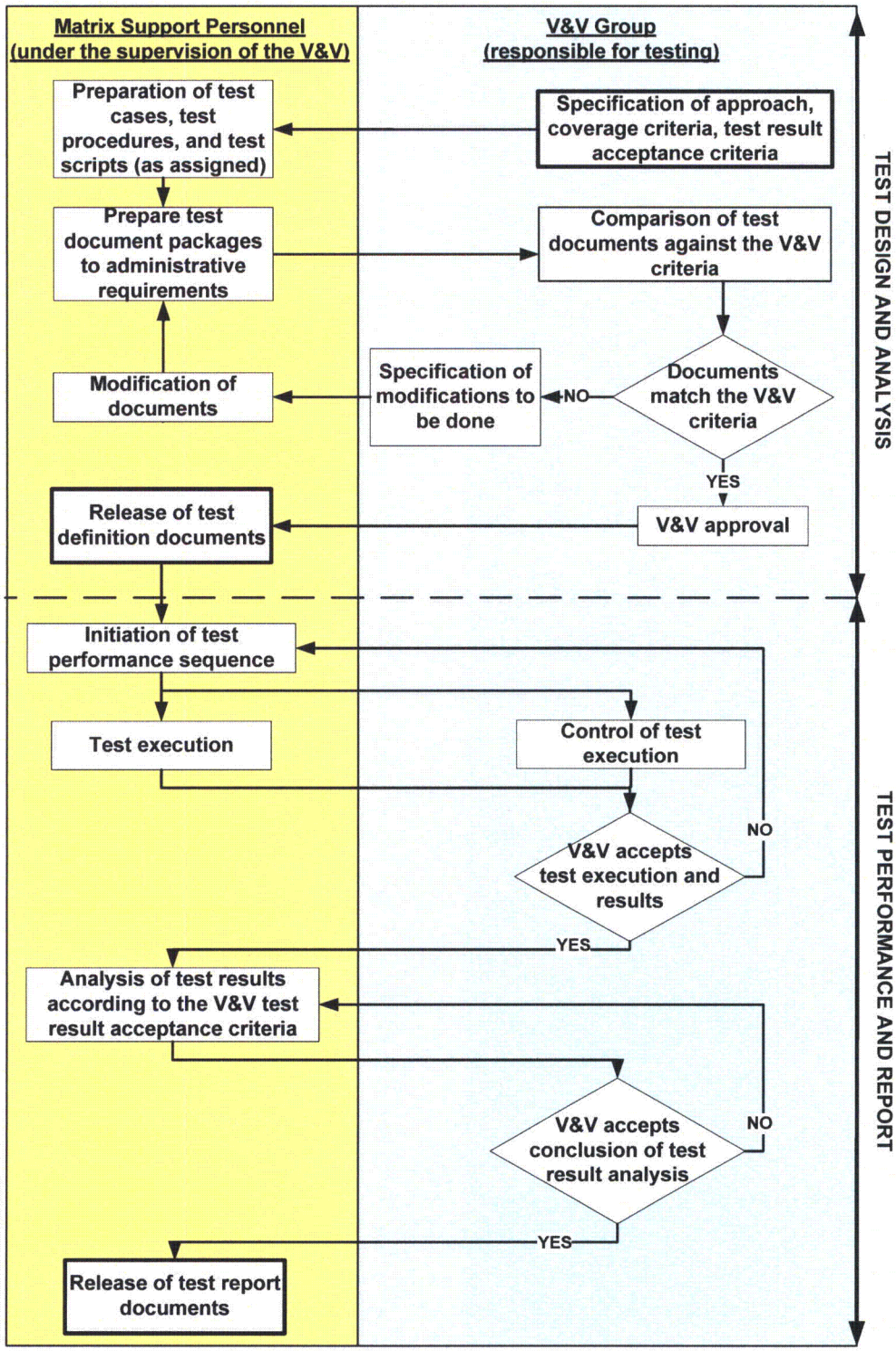
Verification and validation processes provide an objective assessment of software products and processes throughout the software life cycle. This assessment demonstrates whether the software requirements and system requirements allocated to software, are correct, complete, accurate, consistent, and testable. 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 Figure 3-1 and Figure 3-2.

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 direction 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 shown in Figure 11-1 ensure that the independence of validation testing is maintained.

The Software Verification and Validation Plan also conforms to the guidance in IEEE Std 7-4.3.2-2003 clause 5.3.3, as endorsed by Regulatory Guide 1.152.

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



11.3 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 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 also conforms to the guidance in IEEE Std 7-4.3.2-2003 clause 5.3.4, as endorsed by Regulatory Guide 1.152.

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 complexity of the project. This review is performed to ensure effective coordination of Software Safety Plan and Software Quality Assurance Plan activities with the Software Verification and Validation Plan activities. The Verification and Validation group manager is responsible for approving and implementing the Software Verification and Validation Plan.

11.4 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

11.5 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 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.

11.6 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 a SIL level, as described in Section 4.4.

11.7 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 any NRC approved software simulation test tool. The Project Manager arranges and verifies the training and access to these tools.

11.8 Responsibilities

A lead Verification and Validation person 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 from the Software Verification and Validation Plan are made to qualified personnel using the project master schedule.

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 SysRS and the traceability analysis of the SysRS 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 Engineering 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.

The Verification and Validation group is responsible for preparation of the Application Software Integration and System test documents and performance of the validation testing. 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 direction of the Verification and Validation group. These support personnel may not develop software test documents for design work they prepared.

The Verification and Validation group is responsible for correcting errors in validation test documents. The Verification and Validation group reports on anomalies identified as well as the resolution of the anomalies, as specified in IEEE Std 1012-1998.

11.9 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 and the TELEPERM XS Application Software development process, 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. The basis for the AREVA NP approach to software safety analyses is described in Section 10.3.

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

11.9.1 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) – Prepares and verifies that the SILs contained in the Criticality Analysis 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 the interface between Hardware Design Solutions and Software Design Solutions for correctness, accuracy, and completeness of the concept requirement allocation to hardware requirements, software requirements, or both.
- Traceability Analysis (SIL 4, 3) - Develops a preliminary Software Requirements Traceability Matrix using the requirements from the SysRS. 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) - As described in Section 10.0, the software safety analysis activities satisfy the requirements for a software hazards analysis.
- 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.
- Concept Phase Verification and Validation Activity Summary Report (SIL 4, 3, 2) - A summary report of the tasks completed during the Concept Verification and Validation Activity, including the associated metrics.

11.9.2 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) - Performs analysis of the requirements tracing between the SysRS and the SRS to ensure correctness, consistency, accuracy, and completeness. The Software Requirements 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.
- Software Requirements 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) – Updates the Criticality Analysis and verifies that any required changes from the Concept Phase Verification and Validation Activity have been implemented.

- Application Software System and Acceptance Test Plan Generation (SIL 4, 3, 2) – The Application Software System and Acceptance Test Plan (including FAT) describes the system and acceptance test objectives; the features to be tested, test approach, coverage criteria, 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. The Application Software System and Acceptance Test Plan includes the hardware and software supplied by AREVA NP for the total system, from the cabinet input terminals to the output terminals, and peripheral items such as the TELEPERM XS Service Unit, TELEPERM XS Gateway, and TELEPERM XS QDS. System and Acceptance Testing validates that the software and system requirements are correctly implemented utilizing the fully integrated system. The software encompasses system software, including operating system software, communication software, Runtime Environment software, and peripheral software such as Service Monitor Server application and Gateway Historical Application, as well as project-specific Application Software, Graphic Service Monitor screens (if applicable), and TELEPERM XS QDS (if applicable). The test plan also describes what portions of the Application Software System and Acceptance Testing are to be performed with a NRC approved simulation test tool or in the test field.
- 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) - As described in Section 10.0, the software safety analysis activities satisfy the requirements for a software hazards analysis.
- 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 been determined. Verifies that these considerations are contained in the system requirements.
- Requirements Phase Verification and Validation Activity Summary Report (SIL 4, 3, 2) - A summary report of the tasks completed during the Requirements Verification and Validation Activity, including the associated metrics.

11.9.3 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) - 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. All TELEPERM XS Changeable Parameters shall be verified against the appropriate design input documents as part of the software design evaluation.
- Software Design 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) – Updates the Criticality Analysis and verifies that any required changes from the Requirements Phase Verification and Validation Activity have been implemented.
- Application Software Integration Test Plan Generation (SIL 4, 3, 2) –The Application Software Integration Test Plan describes the software testing objectives; features to be tested, test approach, coverage criteria 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. The test plan also describes whether the Application Software Integration Testing is to be performed with a NRC approved simulation test tool or in the test field.
- Application Software Integration Test Specification Generation (SIL 4, 3, 2) - The Application Software Integration Test Specification addresses both test design and test cases. The test specifications reflect whether the Application Software Integration Testing is to be performed with a NRC approved simulation test tool or in the test field.
- Application Software Integration Test Procedure Generation (SIL 4, 3, 2) - Test Procedures are developed to meet the test objectives. The test specifications reflect whether the Application Software Integration Testing is to be performed with a NRC approved simulation test tool or in the test field.
- Hazard Analysis (SIL 4, 3) - As described in Section 10.0, the software safety analysis activities satisfy the requirements for a software hazards analysis.
- 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.

- Design Verification and Validation Activity Summary Report (SIL 4, 3) - A summary report of the tasks completed during the Design Verification and Validation Activity, including the associated metrics.

11.9.4 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) - 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. Verifies that there is a valid relationship between the Application Software Integration Test Procedures and the Application Software Integration Test Plan, Application Software Integration Test Design and Test Case Specification back to the SDD, if Application Software Integration Testing is performed with a NRC approved simulation tool. Verifies that there is a valid relationship between the Application Software System and Acceptance Test Procedures and the Application Software System and Acceptance Test Plan, Application Software System and Acceptance Test Design and Test Case Specification back to the SysRS and SRS, if portions of the Application Software System and Acceptance Testing are performed with a NRC approved simulation tool.
- Source Code and Source Code Documentation Evaluation (SIL 4, 3, 2) - Verifies the following items: Changeable Parameters are verified to ensure that the

appropriate parameters are built into the Application Software Code correctly and within the defined ranges as specified in the associated design documents, SPACE Function Diagrams are compared to the ID Coding Concept document and the Software Design Description to ensure they were prepared correctly, the SPACE Diagrams are further compared to the output of the **scanmic** tool to ensure that the correct revisions were used to assemble the Application Software Release, the Software Configuration Items List and Software Generation and Download procedure are referenced to ensure that approved versions of the code generators were used, and the Code Configuration document is reviewed to ensure that only qualified TELEPERM XS System Software components are used in the executables.

- Software Implementation 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.
- Criticality Analysis (SIL 4, 3, 2) – Updates the Criticality Analysis and verifies that any required changes from the Design Phase Verification and Validation Activity have been implemented.
- Application Software System and Acceptance Test Design Specification Generation (SIL 4, 3, 2) - The System and Acceptance Test Design Specification describes the software features to be tested, approach refinements, and pass/fail criteria.
- Application Software System and Acceptance Test Design and Test Case Specification Generation (SIL 4, 3, 2) - The System and Acceptance Test Design and Test Case Specification identifies the test items, input and output specifications, environmental needs, special procedural requirements, and intercase dependencies.

- Application Software System and Acceptance Test Procedure Generation (SIL 4, 3, 2) – The System and Acceptance Test Procedures describe the test cases, test equipment, prerequisites, test setup, and detailed test procedure steps.
- Application Software Integration Testing (SIL 4, 3) – Validates project-specific Application Software design by executing the Software Integration Testing described in Section 13.3, if Application Software Integration Testing is performed with a NRC approved simulation tool.
- Application Software Integration Test Report and Test Incident Report Verification (SIL 4, 3) - The Application Software Integration Test Report documents that the Application Software satisfies the test acceptance criteria as defined in the Test Specification, if Application Software Integration Testing is performed with a NRC approved simulation test tool.
- Application Software System and Acceptance Testing (SIL 4, 3, 2) – Validates project-specific system design by performing the Application Software System and Acceptance Test Plan described in Section 13.0. This testing is performed during this phase if portions of the Application Software System and Acceptance Testing is performed with a NRC approved simulation tool. The remaining portions of the Application Software System and Acceptance Testing are performed in the test field, as described in Section 11.9.5.
- Hazard Analysis (SIL 4, 3) - As described in Section 10.0, the software safety analysis activities satisfy the requirements for a software hazards analysis.
- 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.
- Implementation Phase Verification and Validation Activity Summary Report (SIL 4, 3) - A summary report of the tasks completed during the Implementation Verification and Validation Activity, including the associated metrics.

11.9.5 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 SysRS are correctly implemented into the fully integrated system and validate project-specific system performance. The objective 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.

- Traceability Analysis (SIL 4, 3) - Verifies that there is a valid relationship between the Application Software System and Acceptance Test Procedures and the Application Software System and Acceptance Test Plan, Application Software System and Acceptance Test Design and Test Case Specification back to the SysRS and SRS for portions of the Application Software System and Acceptance Testing performed in the test field. Verifies that there is a valid relationship between the Application Software Integration Test Procedures and the Application Software Integration Test Plan, Application Software Integration Test Design and Test Case Specification back to the SDD, if Application Software Integration Testing is performed in the test field.
- Application Software Integration Testing (SIL 4, 3) – Validates project-specific Application Software design by performance the Software Test Plan described in

Section 13.0, if Application Software Integration Testing is performed in the test field.

- Application Software Integration Test Report and Test Incident Report Verification (SIL 4, 3) - The Application Software Integration Test Report documents that the Application Software satisfies the test acceptance criteria as defined in the Test Specification, if Application Software Integration Testing is performed in the test field.
- Application Software System and Acceptance Testing (SIL 4, 3, 2) – Validates project-specific system design by performing the Application Software System and Acceptance Test Plan described in Section 13.0.
- Application Software System and Acceptance Test Verification (SIL 4, 3, 2) - Verifies the Application Software System and Acceptance 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 System and Acceptance Test Plan and System and Acceptance Test Procedures. Verifies the correct versions of software were used in the Application Software System and Acceptance Test.
- Hazard Analysis (SIL 4, 3) - As described in Section 10.0, the software safety analysis activities satisfy the requirements for a software hazards analysis.
- 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 Software and Data 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.
- Test Phase Verification and Validation Activity Summary Report (SIL 4, 3, 2) - A summary report of the tasks completed during the Test Verification and Validation Activity, including the associated metrics.

- Software Verification and Validation Final Report (SIL 4, 3, 2) - A report that summarizes the Verification and Validation activities, tasks, and results. The report shall include the status and resolution of all software-related Open Items. The report shall provide an assessment of overall software quality and any recommendations.

11.9.6 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.

11.9.7 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.

11.10 Independent Testing and Validation

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

11.11 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.

11.12 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 how these metrics support the verification and validation objectives. These metrics are also used to conform to the requirements in IEEE Std 7-4.3.2-2003 clause 5.3.1.1, as endorsed by Regulatory Guide 1.152. In addition, the Software Verification and Validation Plan describes the metrics used to measure the effectiveness of the verification and validation effort.

11.12.1 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.

11.12.2 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
- Number of test anomalies discovered during independent verification and validation testing

11.13 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. At a minimum, each verification and validation report will contain the following information:

- Software verification and validation activity identification.
- Background information of standards, and source documents with revisions.
- List of tasks performed and synopsis of the review,
- Results of the verification or validation activity.
- Confirmation that the plan was followed or summary of deviations and their resolution
- Open Items found
- Summary of Open Items Issued and Status
- Summary of Review Results
- Metrics Analysis

- Risk Analysis
- Recommendations

The descriptions of Open Items identified in the report will include as a minimum the following information:

- Date Open Item was initiated
- Open Item number
- Description of the issue involved
- Document(s) of concern
- Assigned priority based on risk analysis

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

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 60 - 67) 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, developing and maintaining the Configuration Item list, controlling the implementation of and changes to software and software documentation, 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 36), as endorsed in Regulatory Guide 1.169 (Reference 10). The Software Configuration Management Plan also conforms to the guidance in IEEE Std 7-4.3.2-2003 clause 5.3.5, as endorsed by Regulatory Guide 1.152.

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 implementing procedure lists the typical 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 12.3 of this manual. The software configuration control begins in the first phase of the project as described in Section 4.0 and continues until the system is released to the customer. Section 2.0 defines key terms.

12.2 Management

The Technical Manager has overall responsibility for the implementation of the Software Configuration Management Plan. The organization is as described in Section 3.2.

The Software Supervisor is assigned specific responsibilities for project configuration management activities, as described in Section 12.2.1. 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 Technical Manager have final approval of change requests.

The Software Librarian is designated in writing and is responsible for the activities described in Section 12.3.3.

12.2.1 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

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-specific Software Configuration Management 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 specific responsibilities 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. The Software Supervisor releases the final version of the TELEPERM XS Application Software to the customer for installation and commissioning in the plant after successful completion of the software validation tests and the system tests (including FAT),.

Personnel assigned to work on TELEPERM XS Application 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 Engineering Tool or any NRC approved simulation test tool) used on each release of the Application Software are controlled and recorded.

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.

12.2.2 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.

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 Engineering 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 Engineering Tool. The Application Software is a direct outcome of this design process; it is not developed separately.

The Application Software Configuration Control Board functions as the low-level Configuration Control Board described in IEEE Std 1042-1987. The purpose of the Application Software Configuration Control Board is to control major issues such as system function and configuration of the software as a whole.

An Application Software Configuration Control Board is convened to address the following conditions:

- Application Software changes for migrations to a newer version of System Software (i.e., changes to the first digit of the revision identifier defined in Section 4.7.1),
- Proposed changes that make significant modifications to the existing Application Software (i.e., changes to the second digit of the revision identifier defined in Section 4.7.1),
- Proposed changes that require a new Application Software release, or
- Any Application Software change made during the Test Phase.

The Configuration Control Board is not used when making other minor modifications to the Application Software documentation that do not affect a Software Release. This approach is acceptable because these changes (i.e., input/output format changes, clarifications, correction of typos, etc.) are limited by the existing functional requirements. All such changes are reviewed and approved via the Open Item process described in Section 3.5.2. The Open Item process provides a sufficient level of review with the appropriate organizations for minor changes.

The Project Engineer serves as chairman of the Application Software Configuration Control Board. The Project Manager, Technical Manager, Software Supervisor, Lead Software Designer, and the Verification and Validation Supervisor serve as board members. As needed, representatives from other organization may be represented on the board. Participation by QA is typically included to ensure compliance with project requirements.

The Application Software Configuration Control Board has the authority to approve or disapprove configuration change requests. The board also has the responsibility for defining changes to the software configuration items and baselines.

12.3 Software Configuration Management Process Controls

The Software Configuration Management Plan implementing procedure 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. The software configuration management process controls described in the Sections 12.3.1 through 12.3.5 are used for TELEPERM XS projects.

12.3.1 Protection of Configuration Items

Configuration control procedures are used to protect configuration items. The procedures describe 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. The implementing procedures also 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 software items are controlled as configuration items:

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

The following list is the minimum set of software documents controlled as configuration items:

- System Architecture
- Software Requirements Specification
- Software Design Description
- Code Document (SPACE Diagrams)

The design and configuration documentation is stored as a quality record in the AREVA NP Records Management System, in accordance with the AREVA NP QA program. This documentation includes TELEPERM XS System Software documentation.

Section 12.3.3 lists the software objects that are stored in the software library.

12.3.2 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 Microsoft® Excel, 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 configuration management baseline and maintained under the configuration controls of the Software Configuration Management Plan.

12.3.3 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 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.

12.3.4 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.

12.3.5 Application Software Release

The Software Supervisor releases the TELEPERM XS Application Software for validation testing after completion of the design process and any engineering debugging tests in the simulation environment. The Software Supervisor releases the final version of the TELEPERM XS Application Software, System Software, and associated documentation to the customer for installation and commissioning in the plant after successful completion of the software validation tests described in Section 13.0.

12.4 Software Configuration Management Implementation Activities

The software configuration management implementation activities described in the Sections 12.4.1 through 12.4.7 are performed for TELEPERM XS projects.

12.4.1 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 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.

12.4.2 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.

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 design 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.

12.4.3 Change Management

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 3.5.2.

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.

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.

12.4.4 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 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 Engineering 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 Engineering Tool configuration is checked and documented with each Software Release created by the steps in this document.

12.4.5 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.

12.4.6 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 3.5.2.

Problem reports that require corrective action involving redesign invoke the AREVA NP Corrective Action Program. Section 3.5 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 3.5.2. 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 3.5.2.

12.4.7 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.

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.

13.0 SOFTWARE TEST PLAN

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

13.1 Purpose

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 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 Application Software System and Acceptance Testing, including FAT.

The generic 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 Integration, System, and Acceptance tests of the Application Software 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:

- 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 software and data security features.

13.2 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).

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.

13.2.1 Component Testing

Component testing requirements for TELEPERM XS Systems are satisfied by the generic qualification type testing of TELEPERM XS hardware and software modules (including Function Blocks) described in Section 4.1.

13.2.2 Integration Testing

Integration testing for a TELEPERM XS System occurs in three stages. TELEPERM XS Platform Integration Testing occurs during the generic qualification process. The purpose of the generic system qualification tests is to validate that the TELEPERM XS Hardware, System Software, and representative Application Software operate correctly in an integrated environment. The system test documentation is listed in Section 8.1.1 of the TELEPERM XS Topical Report.

Application Software Integration Testing is performed in accordance with the Software Verification and Validation Plan, as described in Section 11.0 to conform to IEEE Std 1012-1998 validation requirements. The purpose of the testing is to establish that the project-specific Application Software Function Blocks and Function Diagrams have been properly integrated, based on the project-specific system architecture and SDD. The Application Software Integration Testing can be performed in a simulation environment with an approved simulation test tool or it can be performed in the test field as part of the FAT. This testing is described in Section 13.3.

Project-specific Hardware and Software Integration Testing is performed in accordance with the Software Test Plan. The purpose of the testing is to establish that the project-

specific system has been configured correctly, is operating properly, and that the prerequisites for validation testing to conform with IEEE Std 1012-1998 requirements have been met. This testing is described in Section 13.4.

The terms "software modules" and "subsystems" are generally used to describe groups of software objects. For TELEPERM XS Systems this terminology is applied to a collection of function blocks in the SPACE Function 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.

The Application Software Integration Testing performed for a TELEPERM XS System can be considered as structural or white box testing as defined in IEEE Std 610.12, unit testing as understood in the context of IEEE Std 1008, and module testing as mentioned in BTP 7-14.

13.2.3 System and Acceptance Testing

System and Acceptance Testing for a TELEPERM XS System is designed to validate that the integrated hardware and software system meets its original objectives. Application Software System and Acceptance Testing is performed in accordance with the Software Verification and Validation Plan, as described in Section 11.0 to conform to IEEE Std 1012-1998 validation requirements. The purpose of the System and Acceptance Testing is to establish that the project-specific Application Software have been properly designed, based on the project-specific SysRS and SRS. Some aspects of the Application Software System and Acceptance Testing can be performed in a simulation environment with an approved simulation test tool or it can be performed in the test field as part of the FAT. This testing is described in Section 13.5.

The Application Software System and Acceptance Testing performed for a TELEPERM XS System can be considered as functional or black box testing as defined in IEEE Std 610.12.

13.2.4 Acceptance Testing

Acceptance Testing for a TELEPERM XS System is performed to enable the customer to determine whether to accept the system. The Acceptance Testing usually is based on IEEE Std 1012-1998 validation requirements; however, it may include additional performance test requirements specified by the customer. This testing is also described in Section 13.5.

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** TELEPERM XS Hardware and Software (including Function Blocks) Type Tests	**Not Applicable** (based on use of qualified TELEPERM XS hardware and software modules)
Integration Testing	**X** TELEPERM XS Platform Qualification Testing	Application Software Integration Testing **X** (see Section 13.3 and Note 1) Hardware and Software Integration Testing (see Section 13.4)
System Testing	**X** TELEPERM XS Platform Qualification Testing	Application Software System and Acceptance Testing **X** (see Section 13.5 and Note 1)
Acceptance Testing	**Not Applicable**	

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

Note 1 – Application Software Integration and System and Acceptance Test Cases performed in accordance with any NRC approved simulation test tool can be used to conform with IEEE Std 1012-1998 validation requirements.

13.3 Application Software Integration Testing

The Application Software Integration Test Plan is prepared to ensure that the functional requirements of the software design detailed in the SDD are properly implemented in the SPACE project database. The comprehensiveness of the testing effort should ensure that all functionality defined in the SDD is tested. The Application Software Integration Test Plan defines whether the testing is to be performed using a NRC approved simulation test tool or performed in the test field. The Application Software Integration Test Plan ensures that the software design detailed in the SDD is properly implemented in the SPACE project database.

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

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 recognizes that:

> 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.

13.4 Hardware and Software Integration Testing

The purpose of the Hardware and Software Integration Testing is to establish that the project-specific TELEPERM XS System has been configured correctly, is operating properly, and that the prerequisites for validation testing to conform with IEEE Std 1012-1998 requirements have been met. It is performed after manufacturing tests of the cabinet wiring are completed by the manufacturing organization.

The Hardware and Software Integration 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), and
- System behavior in case of failures of the components and system parts.

The Hardware and Software Integration Tests are divided in several phases.

- 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 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).

The Hardware and Software Integration Testing and the Physical Audits described in Section 5.5.2 serve as prerequisites to validation testing in the test field.

13.5 Application Software System and Acceptance Testing

The Application Software System and Acceptance Test scope includes the following:

- System characteristics (e.g., response times),
- System behavior in case of failures of the components and system parts,
- Correct implementation of the system level I&C requirements and verification of the system interfaces, and
- Test of the specified function of the overall system under selected, simulated operation and failure conditions

The Application Software System and Acceptance Test Plan is prepared to ensure that the functional requirements of the SysRS and SRS are validated. The Application Software System and Acceptance Test Plan defines which portions of the testing is to be performed using a NRC approved simulation test tool and which portions are to be performed in the test field. The comprehensiveness of the testing effort should ensure that all testable requirements of the SysRS and SRS are tested. The project-specific Application Software System and Acceptance Tests fulfill the requirement for System and Acceptance testing and conforms to 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 Application Software System and Acceptance Tests validate that the functionality of the system meets the design and customer requirements in the fully integrated system.

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 Application Software System and Acceptance Testing, including FAT.

This testing includes test cases to address the limitations of any NRC approved simulation test tool used for validation testing. If necessary, real components are connected to the system in the test field (e.g. neutron flux sensor), or control room devices and displays.

13.5.1 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 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).

Otherwise, all I&C functions are tested in the test field.

13.5.1.1 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 testing is the validation of the I&C functionality based on the process engineering tasks specified in the SysRS and SRS.

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 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.

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.

13.5.1.2 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,
- 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 Integration 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.

13.5.1.3 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.

13.5.1.4 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.

13.5.1.5 Plant Scenario 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.

13.5.1.6 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.

13.5.1.7 Software and Data Security Tests

The purpose of this test is to validate the software and data security of the TELEPERM XS System, and demonstrate that requirements for ensuring software and data security protection of the TELEPERM XS System are met. The software and data 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.

13.5.1.8 Diverse Actuation System Tests

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

13.6 Validation Test Documents

The Application Software Integration and System and Acceptance Test Plans are prepared for Application Software validation testing. The Application Software Integration Test Plan ensures that all functionality defined in the SDD is tested. The Application Software System and Acceptance Test Plan ensures that the system requirements detailed in the SysRS and SRS are properly implemented in the system design.

The Application Software Integration and System and Acceptance Testing is performed under the direction of the Verification and Validation Group. The Application Software Integration and System and Acceptance 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 Application Software Integration and System and Acceptance 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 the testable system requirements (from the SysRS and SRS) have been tested in either the Application Software Integration Test or the System and Acceptance Test. The Verification and Validation Group also uses the Software Requirements Traceability Matrix to ensure that all functionality defined in the SDD is tested. The Verification and Validation Group independently verifies that the Application Software versions being tested match those listed in the Software Configuration Management documentation. The test documentation identifies the versions of the software being tested and includes the results of the version verification activities.

13.7 Test Reporting

The Verification and Validation Group verifies the Application Software Integration and System and Acceptance 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 Application Software Integration and System and Acceptance Test Plans and Test Procedures. The Verification and Validation Group verifies the correct versions of software were used in the validation testing.

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

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.
- 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 Procedure with no unexpected intermediate results.
- A test Item containing unexpected results may be acceptable and considered 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, the Test Specification, or the Test Procedure are found to be in error (i.e., the results of the test do not match the predicted results described in the Test Procedure).

The Open Items system, as described in Section 3.5.2, 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 3.5.

13.8 Test Tools

Test Scripts are used to simulate plant condition inputs on the TELEPERM XS System and to automate steps in the individual Test Procedures. The Test Scripts are included in the Test Procedure documentation in the exact format (without modification). The electronic version of the Test Scripts are verified prior to use using the authentication tools described in Section 0. The authentication tools are also used to document the Test Script directory by creating CRC checksums for all files in the directory and including the listing as an attachment to the Test Procedures. After testing is completed, the final electronic versions of the Test Scripts are stored in the Software Library, as described in Section 12.3.3.

Commercial software packages (e.g., Microsoft® Excel), which may use macros or scripts, are used to support the review of test results. These software packages are not credited as the method for performing final review of test results unless they have been verified and validated as acceptable for use. The tool macros are verified daily both before and after use during testing. The macro is validated by test personnel prior to testing activities by consecutive manual comparisons of test data using special test

cases. The test case identifies the expected test results, the results data being evaluated, and the correct output of the Excel macro after execution. This process ensures that the tool macro is performing correctly before use and is reverified after its use. This will eliminate the possibility that the tool macro may introduce errors into the validation test results evaluation process. The results of the verification activities are documented in the test records. If the software tool has not been verified, then the actual test results are manually verified.

Special tools needed for performance of the Application Software System and Acceptance Tests are the ERBUS Test Machine and the data acquisition equipment to generate simulated input signals and records the output signals. Other special tools needed include calibrated low level current sources and fast response recording devices for performing any required nuclear instrumentation and response time tests. The requirements for this equipment and other needed test tools are specified in the corresponding Test Specifications (for software tests) or Test Procedures (for hardware tests). The measurement tolerance (or the minimum precision for measured output of analog signals) is defined in the Test Specifications (for software tests) or Test Procedures (for hardware tests) and used when determining test acceptance criteria in test procedures.

13.9 Software Testing Summary

Component testing requirements for TELEPERM XS Systems are satisfied by the generic qualification type testing of TELEPERM XS hardware and software modules (including Function Blocks).

Integration testing for a TELEPERM XS System occurs in three stages: TELEPERM XS Platform Integration Testing occurs during the generic qualification process, Application Software Integration Testing is performed in accordance with the Software Verification and Validation Plan, and project-specific Hardware and Software Integration Testing is performed as a prerequisite for FAT.

Application Software System and Acceptance Testing for a TELEPERM XS System is performed in accordance with the Software Verification and Validation Plan to validate that the functionality of the system meets the design and customer requirements in the fully integrated system.

Acceptance Testing for a TELEPERM XS System is performed to enable the customer to determine whether to accept the system.

The TELEPERM XS Application Software Test Plan conforms to the requirements of IEEE Std 1012-1998.

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 following plans, which address the BTP 7-14 review guidance:

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

The Software Program Manual is AREVA NP's upper tier requirements document for TELEPERM XS application software development. The statements in the Software Program Manual are considered to be the programmatic requirements necessary to implement the TELEPERM XS application software program.

AREVA NP Operating Instructions will be used to define specific implementations details, as necessary. 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.

The Software Program Manual will be implemented by procedures to produce the documentation that demonstrates that the project-specific implementation activities have been successfully accomplished for each life cycle activity group. These activities are designed to ensure that the:

- Project-specific Application Software is safe,
- System safety requirements have been adequately addressed for each activity group,
- No new hazards have been introduced, and
- Software requirements, design elements, and code elements that can affect safety have been identified, and that all other software requirements, design, and code elements will not adversely affect safety.

The Software Program Manual will be implemented by procedures to produce the project-specific design outputs required by the Software Development Plan and produced in accordance with the Software Management Plan and the Software Quality Plan.

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 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.

15.0 REFERENCES

15.1 U.S. Regulations

1. 10 CFR Part 21, "Reporting of Defects and Noncompliance."
2. 10 CFR 50.65, "Requirements for Monitoring the Effectiveness of Maintenance at Nuclear Power Plants."
3. 10 CFR Part 50, Appendix B, "Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants."
4. 10 CFR 73.54, "Protection of Digital Computer and Communication Systems and Networks."
5. 10 CFR Part 100, "Reactor Site Criteria."

15.2 U.S. Regulatory Guidance

6. Regulatory Guide 1.53, Revision 2, November 2003, "Application of the Single-Failure Criterion to Safety Systems."
7. Regulatory Guide 1.152, Revision 2, January 2006, "Criteria for Digital Computers in Safety Systems of Nuclear Power Plants."
8. Regulatory Guide 1.153, Revision 1, June 1996, "Criteria for Safety Systems."
9. 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."
10. Regulatory Guide 1.169, September 1997, "Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants."
11. Regulatory Guide 1.170, September 1997, "Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants."
12. Regulatory Guide 1.171, September 1997, "Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants."
13. Regulatory Guide 1.172, September 1997, "Software Requirements Specifications for Digital Computer Software Used In Safety Systems of Nuclear Power Plants."
14. Regulatory Guide 1.173, September 1997, "Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants."
15. Regulatory Guide 1.187, November 2000, "Guidance for Implementation of 10 CFR 50.59, Changes, Tests, and Experiments."

16. NUREG-0800, Standard Review Plan (SRP), Section 7.9, "Data Communication Systems," Revision 5.

17. NUREG-0800, SRP, Chapter 7, Branch Technical Position (BTP) 7-14, "Guidance on Software Review for Digital Computer-Based Instrumentation and Control Systems."

18. 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."

19. DI&C-ISG-01, "Task Working Group #1: Cyber Security Interim Staff Guidance," Revision 0.

20. DI&C-ISG-04, "Task Working Group #4: Highly-Integrated Control Rooms—Communications Issues (HICRc) Interim Staff Guidance," Revision 1.

15.3 U.S. Industry Standards

21. ANSI/ASME NQA-1 – 1994, "Quality Assurance Program Requirements for Nuclear Facilities."

22. EPRI TR-102348, Revision 1, "Guideline on Licensing Digital Upgrades."

23. IEEE Std 7-4.3.2-2003, "Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations."

24. IEEE Std 379-2000, "Application of the Single-Failure Criterion to Nuclear Power Generating Station Safety Systems."

25. IEEE Std 352-1987, "Guide for General Principles of Reliability Analysis of Nuclear Power Generating Station Safety Systems."

26. IEEE Std 577-1976, "Requirements for Reliability Analysis in the Design and Operation of Safety Systems for Nuclear Power Generating Stations."

27. IEEE Std 603-1991, "Criteria for Safety Systems in Nuclear Power Plants."

28. IEEE Std 610.12-1990, "Software Engineering Terminology."

29. IEEE Std 730-2002, "Standard for Software Quality Assurance Plans."

30. IEEE Std 828-1990, "Standard for Software Configuration Management Plans."

31. IEEE Std 829-1983, "Standard for Software Test Documentation."

32. IEEE Std 830-1993, "Recommended Practice for Software Requirements Specifications."

33. IEEE Std 1008-1987, "IEEE Standard for Software Unit Testing."

34. IEEE Std 1012-1998, "Standard for Software Verification and Validation."

35. IEEE Std 1028-1997, "Standard for Software Reviews."

36. IEEE Std 1042-1987, "Guide to Software Configuration Management."

37. IEEE Std 1063-2001, "IEEE Standard for User Documentation."

38. IEEE Std 1074-1995, "IEEE Standard for Developing Software Life Cycle Processes."

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

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

41. NEI 96-07, Revision 1, "Guidelines for 10 CFR 50.59 Implementation."

42. NEI 04-04 Revision 1, "Cyber Security Program for Power Reactors."

43. IEC 60880, "Software for Computers in the Safety Systems of Nuclear Power Stations," (1986 version).

44. KTA 3503, "Type Testing of Electrical Modules for the Reactor Protection System," (November 1986 edition).

15.4 Regulatory Review Precedent

45. NRC Regulatory Issue Summary 2002-22, Use of EPRI/NEI Joint Task Force Report, "Guideline On Licensing Digital Upgrades: EPRI TR-102348, Revision 1, NEI 01-01: A Revision of EPRI TR-102348 to Reflect Changes to the 10 CFR 50.59 Rule."

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

47. Letter, Juan Peralta (NRC) to Hans-Joachim Nisslein (AREVA NP), "NRC Inspection Report for AREVA NP GmbH 99901371/2008-201," dated May 7, 2008.

48. Letter, James F. Mallay (Siemens) to Document Control Desk (NRC), "Supporting Documentation for Review of EMF-2110(NP) Revision 1, TELEPERM XS: A Digital Reactor Protection System" NRC:99:037, dated September 1, 1999.

49. Letter, James F. Mallay (Siemens) to Document Control Desk (NRC), "Additional Information in Support of the TELEPERM XS Review" NRC:99:056, dated December 28, 1999.

50. Letter, James F. Mallay (Siemens) to Document Control Desk (NRC), "Additional Information in Support of the TELEPERM XS Review" NRC:00:004, dated January 13, 2000.

51. Letter, Dave Baxter (Duke Energy) to Document Control Desk (NRC), "Oconee Nuclear Station, Units 1, 2, and 3 Supplemental Request for Additional information for License Amendment Request for Reactor Protective

System/Engineered Safeguards Protective System Digital Upgrade, Technical Specification Change (TSC) Number 2007-09, Supplement 13," dated March 6, 2009.

52. Letter, Dave Baxter (Duke Energy) to Document Control Desk (NRC), "Oconee Nuclear Station, Units 1, 2, and 3 Supplemental Request for Additional information for License Amendment Request for Reactor Protective System/Engineered Safeguards Protective System Digital Upgrade, Technical Specification Change (TSC) Number 2007-09, Supplement 16," dated April 30, 2009.

53. Letter, John F. Stang (NRC) to Dave Baxter (Duke Energy), "Audit Report by the Office of Nuclear Reactor Regulation Staff on Oconee Nuclear Station, Units 1, 2, and 3, Design Review Audit at AREVA for Digital Upgrade of the Reactor Protective System And Engineered Safeguards Prototective (sic) System," dated June 22, 2009.

15.5 AREVA NP Documents

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

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

56. Siemens Topical Report EMF-2267(P), Revision 0, "Siemens Power Corporation Methodology Report for Diversity and Defense-in-Depth," September 1999.

57. Siemens Topical Report EMF-2340(P), Revision 0, "Siemens Power Corporation Typical Diversity and Defense-in-Depth Assessment in Accordance with the Methodology of EMF-2267(P)," January 2000.

58. Siemens Topical Report EMF-2341(P), Revision 1, "Generic Strategy for Periodic Surveillance Testing of TELEPERM™ XS Systems in U.S. Nuclear Generating Stations," March 2000.

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

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

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

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

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

64. AREVA NP Administrative Procedure 0503-19, "Transmittal of Deliverable Safety-Related Product Documentation to Customers."

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

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

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

68. AREVA NP GmbH Engineering Procedure FAW TELEPERM XS-1.7, "Information Security."

IEEE Std 1074-1995 Software Life Cycle ↓	AREVA NP Application Life Cycle Phases IEEE Std 1074-1997 Software Life Cycle ↓	Elaboration of the Quotation	Project Start-Up	Basic Design	Detailed Design	Manufacturing	Testing	Installation and Commissioning	Final Documentation	Notes
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									See Software Program Manual Sections 3.3, 4.2, and 4.3.
	A.1.1.2, Perform Estimations	X	X							
3.1.4, Allocate Project Resources	A.1.1.3, Allocate Project Resources	X	X							See Software Program Manual Section 4.5.1.
3.3.4, Define Metrics	A.1.1.4, Define Metrics		X							See Software Program Manual Section 3.5.2, 5.9.2, and 11.12.
3.1.5, Establish Project Environment	A.1.2.1, Plan Evaluations		X							See Software Program Manual Sections 3.3, 4.2, and 4.7.
7.2.3, Plan Configuration Management	A.1.2.2, Plan Configuration Management		X	X						See Software Program Manual Section 12.
4.1.6, Plan System Transition	A.1.2.3, Plan System Transition		X	X						See Software Program Manual Sections 3.5.2 and 12.2.2 for changes to TELEPERM XS System Software. Retirement Phase is outside the scope of the Software Program Manual.
6.1.3, Plan Installation	A.1.2.4, Plan Installation		X							See Software Program Manual Section 7.
7.3.3, Plan Documentation	A.1.2.5, Plan Documentation		X							See Software Program Manual Sections 4.5, 4.7, 5.3, 5.6, 10.3, 11.13, and 13.6.
7.4.3, Plan the Training Program	A.1.2.6, Plan Training		X							AREVA NP staff training is described in Software Program Manual Sections 3.3, 5.13, 10.2, and 11.7. Customer training activities are outside the scope of the Software Program Manual. See Software Program Manual Section 9 for AREVA NP approach to customer training.
3.1.6, Plan Project Management	A.1.2.7, Plan Project Management		X							See Software Program Manual Sections 3 and 4, especially 3.3.2, 3.5, and 4.5.1.
5.3.7, Plan Integration	A.1.2.8, Plan Integration		X							See Software Program Manual Section 6.
	A.1.3.1, Input Information	X	X	X	X					
3.2.5, Manage the Project	A.1.3.2, Manage the Project	X	X	X	X	X	X	X	X	See Software Program Manual Section 3 and also Sections 3.5.2, 5.9.2, and 11.12.
3.3.6, Identify Quality Improvement Needs	A.1.3.3, Identify SLCP Improvement Needs	X							X	See Software Program Manual Sections 5.3 through 5.6 and 11.13.
3.2.6, Retain Records	A.1.3.4, Retain Records			X	X	X	X	X	X	See Software Program Manual Sections 3.1.1, 4.1.4, 5.12, and 12.3.1.
	A.1.3.5, Collect and Analyze Metric Data			X	X	X	X	X	X	
4.1.3, Identify Ideas or Needs	A.2.1.1 Identify Ideas or Needs		X	X	X					See Software Program Manual Section 3.5.2.
4.1.4, Formulate Potential Approaches	A.2.1.2, Formulate Potential Approaches			X	X					See Software Program Manual Sections 3.5.2 and 12.2.2.
4.1.5, Conduct Feasibility Studies	A.2.1.3, Conduct Feasibility Studies			X						See Software Program Manual Sections 3.5.2 and 12.2.2.

	AREVA NP Application Life Cycle Phases	Elaboration of the Quotation	Project Start-Up	Basic Design	Detailed Design	Manufacturing	Testing	Installation and Commissioning	Final Documentation	Notes
IEEE Std 1074-1995 Software Life Cycle ↓	IEEE Std 1074-1997 Software Life Cycle ↓									
4.1.7, Refine and Finalize the Idea or Need	A.2.1.4, Refine and Finalize the Idea or Need			X						See Software Program Manual Sections 3.5.2 and 12.2.2.
4.2.3, Analyze Functions	A.2.2.1, Analyze Functions			X	X					See Software Program Manual Section 4.3.
4.2.4, Develop System Architecture	A.2.2.2, Develop System Architecture			X						See Software Program Manual Sections 4.1 and 4.3.
4.2.5, Decompose System Requirements	A.2.2.3, Decompose System Requirements				X					See Software Program Manual Section 4.3.
	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 Applicable)			X						
	A.2.3.4, Import Software (If Applicable)			X						
5.1.3, Define and Develop Software Requirements	A.3.1.1, Define and Develop Software Requirements			X						See Software Program Manual Sections 4.3, 4.5.2, 4.5.4, 5.4.1, and 11.9.2.
5.1.4, Define Interface Requirements	A.3.1.2, Define Interface Requirements			X						See Software Program Manual Sections 4.3, 4.5.2, 4.5.4, 5.4.1, and 11.9.2.
5.1.5, Prioritize and Integrate Software Requirements	A.3.1.3, Prioritize and Integrate Software Requirements			X						See Software Program Manual Sections 4.3, 4.5.2, 4.5.4, 5.4.1, and 11.9.2.
5.2.3, Perform Architectural Design	A.3.2.1, Perform Architectural Design			X						See Software Program Manual Sections 4.3, 4.5.3, 4.5.5, 5.4.2, and 11.9.3.
5.2.4, Design Data Base (If Applicable)	A.3.2.2, Design Data Base (If Applicable)									Not applicable to TELEPERM XS technology.
5.2.5, Design Interfaces	A.3.2.3, Design Interfaces				X					See Software Program Manual Sections 4.3, 4.5.3, 4.5.5, 5.4.2, and 11.9.3.
5.2.7, Perform Detailed Design	A.3.2.4, Perform Detailed Design				X					See Software Program Manual Sections 4.3, 4.5.3, 4.5.5, 5.4.3, and 11.9.3.
5.3.4, Create Source Code; 5.3.5, Create Object Code	A.3.3.1, Create Executable Code				X					See Software Program Manual Sections 4.2, 4.3, 4.5.6, 5.4.3, and 11.9.4. Note that topics of source code comments are not applicable to code created with SPACE Tool.
5.3.6, Create Operating Documentation	A.3.3.2, Create Operating Documentation				X	X				See Software Program Manual Sections 4.2, 4.3, 4.5.6, 5.4.3, and 11.9.4.
5.3.8, Perform Integration	A.3.3.3 Perform Integration				X		X			See Software Program Manual Sections 6, 13.3, and 13.4.
6.1.4, Distribute Software	A.4.1.1, Distribute Software				X		X	X	X	See Software Program Manual Section 12.3.5.
6.1.5, Install Software	A.4.1.2, Install Software				X		X	X		See Software Program Manual Section 8.
6.1.6, Accept Software in Operational Environment	A.4.1.3, Accept Software in Operational Environment				X		X	X		See Software Program Manual Sections 11.9.5 and 13.5.

	AREVA NP Application Life Cycle Phases	Elaboration of the Quotation	Project Start-Up	Basic Design	Detailed Design	Manufacturing	Testing	Installation and Commissioning	Final Documentation	Notes
IEEE Std 1074-1995 Software Life Cycle ↓	IEEE Std 1074-1997 Software Life Cycle ↓									
6.2.3, Operate the System	A.4.2.1, Operate the System									Operation and Maintenance Phases are outside the scope of the Software Program Manual. See Software Program Manual Section 8 for AREVA NP vendor problem reporting responsibilities.
6.2.4 Provide Technical Assistance and Consulting	A.4.2.2, Provide Technical Assistance and Consulting									Operation and Maintenance Phases are outside the scope of the Software Program Manual. See Software Program Manual Section 8 for AREVA NP technical support responsibilities.
6.2.5, Maintain Support Request Log	A.4.2.3, Maintain Support Request Log									Operation and Maintenance Phases are outside the scope of the Software Program Manual. See Software Program Manual Section 8 for AREVA NP technical support responsibilities.
3.3.6, Identify Quality Improvement Needs	A.4.3.1, Identify Software Improvement Needs			X	X	X	X			See Software Program Manual Sections 5.3 through 5.6 and 11.13.
3.2.7, Implement Problem Reporting Methods	A.4.3.2, Implement Problem Reporting Methods			X	X	X	X	X	X	See Software Program Manual Sections 3.5 and 8.2.
6.3.3, Reapply SLC	A.4.3.3, Reapply SLC									Maintenance Phase is outside the scope of the Software Program Manual. See Software Program Manual Section 8 for AREVA NP technical support responsibilities.
6.4.3, Notify User	A.4.4.1, Notify User									Maintenance Phase is outside the scope of the Software Program Manual. See Software Program Manual Section 8 for AREVA NP technical support responsibilities.
6.4.4, Conduct Parallel Operations (If Applicable)	A.4.4.2, Conduct Parallel Operations (If Applicable)									Not applicable to TELEPERM XS technology.
6.4.5, Retire System	A.4.4.3, Retire System									Retirement Phase is outside the scope of the Software Program Manual. See Software Program Manual Section 8 for AREVA NP technical support responsibilities.
	A.5.1.1, Conduct Reviews			X	X	X	X			
	A.5.1.2, Create Traceability Matrix			X	X	X	X		X	
	A.5.1.3, Conduct Audits			X	X	X	X			
	A.5.1.4, Develop Test Procedures				X		X			
5.3.3, Create Test Data	A.5.1.5, Create Test Data,				X		X			See Software Program Manual Sections 11.9.3, 11.9.4, 11.9.5 and13.6.
	A.5.1.6, Execute Tests				X		X			
	A.5.1.7, Report Evaluation Results				X		X			

IEEE Std 1074-1995 Software Life Cycle ↓	AREVA NP Application Life Cycle Phases IEEE Std 1074-1997 Software Life Cycle ↓	Elaboration of the Quotation	Project Start-Up	Basic Design	Detailed Design	Manufacturing	Testing	Installation and Commissioning	Final Documentation	Notes
7.2.4, Develop Configuration Identification	A.5.2.1, Develop Configuration Identification			X						See Software Program Manual Section 12.
7.2.5, Perform Configuration Control	A.5.2.2, Perform Configuration Control			X	X	X	X	X	X	See Software Program Manual Section 12.
7.2.6, Perform Status Accounting	A.5.2.3, Perform Status Accounting			X	X		X		X	See Software Program Manual Section 12.4.7.
7.3.4, Implement Documentation	A.5.3.1, Implement Documentation			X	X	X	X	X	X	See Software Program Manual Sections 4.5, 4.7, 5.3, 5.6, 10.3, 11.13, and 13.6.
7.3.5, Produce and Distribute Documentation	A.5.3.2, Produce and Distribute Documentation			X	X	X	X	X	X	See Software Program Manual Sections 4.5, 4.7, 5.3, 5.6, 10.3, 11.13, and 13.6.
7.4.4, Develop Training Materials	A.5.4.1, Develop Training Materials		X							AREVA NP staff training is described in Software Program Manual Sections 3.3, 5.13, 10.2, and 11.7. Customer training activities are outside the scope of the Software Program Manual. See Software Program Manual Section 9 for AREVA NP approach to customer training.
7.4.5, Validate the Training Program	A.5.4.2, Validate the Training Program									AREVA NP staff training is described in Software Program Manual Sections 3.3, 5.13, 10.2, and 11.7. Customer training activities are outside the scope of the Software Program Manual. See Software Program Manual Section 9 for AREVA NP approach to customer training.
7.4.6, Implement the Training Program	A.5.4.3, Implement the Training Program		X	X	X	X	X	X		AREVA NP staff training is described in Software Program Manual Sections 3.3, 5.13, 10.2, and 11.7. Customer training activities are outside the scope of the Software Program Manual. See Software Program Manual Section 9 for AREVA NP approach to customer training.
3.2.3, Analyze Risks		X	X							See Software Program Manual Section 5.14.
3.2.4, Perform Contingency Planning		X	X							See Software Program Manual Section 5.14.
3.3.3, Plan Software Quality Management			X							See Software Program Manual Section 5.14.
3.3.5, Manage Software Quality				X	X		X		X	See Software Program Manual Sections 5.3 through 5.6.
5.2.6, Design or Select Algorithms					X					See Software Program Manual Sections 4.3, 4.5.3, 4.5.5, 5.4.2, and 11.9.3.
7.1.3, Plan Verification and Validation			X							See Software Program Manual Section 11.
7.1.4, Execute Verification and Validation Tasks				X	X		X			See Software Program Manual Sections 11.9 and 13. Also see Sections 3.5.4 and 3.5.5.
7.1.5, Collect and Analyze Metric Data				X	X		X			See Software Program Manual Section 11.12.
7.1.6 Plan Testing (Verification and Validation)				X	X					See Software Program Manual Sections 11.9.2, 11.9.3, and 13.

	AREVA NP Application Life Cycle Phases	Elaboration of the Quotation	Project Start-Up	Basic Design	Detailed Design	Manufacturing	Testing	Installation and Commissioning	Final Documentation	Notes
IEEE Std 1074-1995 Software Life Cycle ↓	IEEE Std 1074-1997 Software Life Cycle ↓									
7.1.7, Develop Test Requirements (Verification and Validation)					X		X			See Software Program Manual Sections 11.9.3, 11.9.4, 11.9.5, and 13.
7.1.8, Execute the Tests (Verification and Validation)					X		X			See Software Program Manual Sections 11.9.5 and 13.

Appendix B – Historical Information on SPACE Engineering 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

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 Engineering 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 Engineering 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.

Application software is developed using the SPACE Engineering 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 Engineering Tool. Logical 'software integration' occurs at this stage. The project-specific TELEPERM XS system is developed from qualified hardware and software modules using the approved development tools.

Physical software integration occurs during the FAT stage, when the Application Software is loaded on the TELEPERM XS processors. The project-specific Application Software System and Acceptance 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 Engineering 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 SPACE Engineering 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 Engineering 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 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 Engineering 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 project 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 Engineering Tool code generator. The reconstructed specifications from the code retranslation are compared with specification data contained in the SPACE project 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 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 GmbH used this feature for a period of time then replaced the capability with a separate tool (**rediff**). AREVA NP GmbH 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 AREVA NP GmbH internal QA procedures.

B.5 Experience with RETRANS

As noted in TELEPERM XS Topical Report Section 2.4.3.3.3, AREVA NP GmbH (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 GmbH 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 model of the SPACE project 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 project 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 Engineering Tool 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

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 database 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 GmbH using RETRANS.

Appendix C – TELEPERM XS Software and Data Security Features

C.1 Introduction

The TELEPERM XS system provides a robust and multi-layered approach to addressing the issue of software and data security. The standard TELEPERM XS System is described in the TELEPERM XS Topical Report (Reference 55). The report describes both the design features and development process for the TELEPERM XS hardware platform and operating system. The development process for TELEPERM XS Application Software for U.S. projects is described in this Topical Report. Software and data security for the TELEPERM XS System is addressed through robust design features of the TELEPERM XS hardware platform and System Software (including the Function Block library), process controls for the System Software developed by AREVA NP GmbH in Germany, and process controls for Application Software developed by AREVA NP Inc. in the U.S.

The standard TELEPERM XS platform (hardware and operating system) was designed several years prior to the issuance of Revision 2 of Regulatory Guide 1.152 (Reference 7). As such, some elements of the generic system qualification are not explicitly addressed as software and data security activities in the TELEPERM XS Topical Report and the associated NRC Safety Evaluation Report (Reference 46). However, it must also be recognized that aspects of the TELEPERM XS platform design that address the nuclear safety aspects of communication independence, safety to non-safety system isolation, interference-free communication, and access control are equally applicable to the software and data security attributes of confidentiality, integrity, and availability.

The TELEPERM XS System Software development process controls related to software and data security are briefly mentioned in the TELEPERM XS Topical Report but not mentioned in the NRC Safety Evaluation Report. A more complete description of the development controls for both System and Application Software is provided in Section C.5.

C.2 Software and Data Security Program Objectives

The TELEPERM XS System conforms to the software and data security objectives of ensuring system confidentiality, integrity, and availability (as expressed in Reference 4). Confidentiality, in the context of software and data security, is generally understood as ensuring that information is accessible only to those authorized to have access. Integrity, in the context of software and data security, is generally understood as ensuring that an unauthorized party cannot modify a system setup or the information saved in a system or transferred over a network. Availability, in the context of software and data security, is generally understood as ensuring that a legitimate user can maintain access to information, a system, or a network.

C.3 TELEPERM XS System Operating Principles

The TELEPERM XS System is designed with specific operating principles that provide interference free communication and communication independence. These same features also limit the consequences of unauthorized software and data events on a TELEPERM XS System.

The TELEPERM XS System is designed for strictly cyclic signal processing. No process interrupts are used. This design feature increases the predictability of the software and ensures that operation is unaffected by plant events, as signaled through the data processed. Watchdog timers are used to monitor processor operation.

The TELEPERM XS System is designed as an asynchronous operating system. The TELEPERM XS System does not utilize a real-time clock to synchronize redundant processors. This design feature eliminates synchronization failure modes and enhances system reliability.

The TELEPERM XS System uses only static memory allocation. Each variable in the Application Software has a permanent dedicated place in memory defined at the time of

code generation. This feature eliminates failure modes associated with memory conflicts caused by dynamic memory allocation.

The TELEPERM XS System uses a unique identification (ID) for each TELEPERM XS computer, configuration data for all input/output drivers, and the complete lists of all messages to be sent or received by the TELEPERM XS computers. The predefined messages and communication channels are used by the TELEPERM XS processors to filter out unrecognized messages. Only expected messages addressed to the particular TELEPERM XS processor are accepted. The predefined messages and communication channels are assigned during Application Software code generation, which is done automatically using the TELEPERM XS SPACE Engineering Tool automatic code generators.

The TELEPERM XS communication protocol does not use acknowledges by the receiver. Consequently, it is deterministically excluded that the receiver of a message can have any influence on the sender's operation. Likewise, the sender cannot influence the operation of the receiver. The TELEPERM XS safety processors operate independently of the communication processing and make fail safe decisions in the case of lost messages or interrupted communication.

C.4 TELEPERM XS System Features that Minimize Software and Data Security Vulnerabilities

The TELEPERM XS System has a number of features that minimize or eliminate software and data security vulnerabilities.

The TELEPERM XS Application Software is plant-specific and developed with NRC-approved tools. The TELEPERM XS System Software and Function Block library are plant-independent, pre-developed, NRC-approved software qualified for safety applications.

The TELEPERM XS System Software, Function Block library, and Application Software are custom and proprietary software. The use of the TELEPERM XS System Software is limited to nuclear applications. Consequently, the software does not have the same software and data security vulnerabilities as open-access industrial software or widely distributed public software.

The TELEPERM XS Safety System does not use Transmission Control Protocol/Internet Protocol (TCP/IP) software or message routing services, which are software features that are commonly involved in unauthorized software modifications and data events. As such, the TELEPERM XS System Software strategy eliminates important transmission vectors for unauthorized external software and data events.

The TELEPERM XS System is designed to execute program code from Flash Erasable Programmable Read Only Memory (FEPROM). The program code on the FEPROM can only be changed from the TELEPERM XS Service Unit during the boot load process using the maintenance laptop. The TELEPERM XS System is designed without the use of external data storage media. Consequently, the TELEPERM XS System has no memory storage media where malevolent code can hide or replicate. These design features eliminate the vulnerability to unauthorized software and data events caused by malevolent code.

The TELEPERM XS software development processes contains multiple layers of control to ensure that the system design products (hardware and software) do not contain undocumented code, and other unauthorized or undocumented functions or applications. These features provide software and data security protection from unauthorized modifications.

The TELEPERM XS System uses a combination of personnel access controls, administrative controls, and design features (including hardware and software elements with appropriate alarms) to protect the safety-related software from unauthorized alteration. The design features are a combination of TELEPERM XS platform/system

design features (e.g., self surveillance and CRC tests) and application design features (e.g., physical key switches for control of safety function processor modes). These features provide software and data security protection from unauthorized modifications.

C.4.1 TELEPERM XS Software Design Features Summary

The TELEPERM XS design features that provide communication independence, safety to non-safety system isolation, and interference-free communication also address the software and data security attributes of confidentiality, integrity, and availability. The TELEPERM XS proprietary software does not have the same software and data security vulnerabilities as open-access industrial software or widely distributed public software. The TELEPERM XS System does not use software features that are commonly involved in software and data events, thereby eliminating important transmission vectors for unauthorized external software and data events. The TELEPERM XS design features eliminate the vulnerability to unauthorized software modifications and data events caused by malevolent code and ensure that any modified code is promptly identified and isolated. The TELEPERM XS design features eliminate the vulnerability to network data storms. The TELEPERM XS message transfer protocol, when coupled with other design features, eliminates additional software and data security concerns.

C.4.1.1 TELEPERM XS Computer Message Transfer Protocols

Communication between TELEPERM XS computers is designed with logical point-to-point connections. Separate logical MicroNET communication channels are provided for each message. The TELEPERM XS System is designed for strictly cyclic data transfer via network connections, using a predefined package size and a constant bus load, with checksum and message age monitoring. These features are defined by the SPACE Engineering Tool Code Generators at the time of code generation. The TELEPERM XS System is designed to operate with a fixed cycle time. Each

TELEPERM XS computer reads input data and writes output data once per cycle. This design feature eliminates the vulnerability to network data storms.

The message size is determined individually for each message at the time of code generation by the TELEPERM XS automated code generation tool. The TELEPERM XS design uses a standard message frame that consists of a standard header and a fixed size data section. Message header consists of the following information:

Security-Related Information Withheld in Accordance with 10 CFR 2.390

The integrity of all messages received from other TELEPERM XS computers is checked as part of the cyclic operation. The following checks are performed:

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Service commands sent by the TELEPERM XS Service Unit in service messages are addressed to an individual TELEPERM XS computer. Service commands from the

TELEPERM XS Service Unit are accepted for execution by a TELEPERM XS computer only under the following conditions:

- Message is valid and addressed to its own processing module (correct ID),
- Processing of a previous service message is complete and the TELEPERM XS computer is ready to accept a new service message,
- Sequence check detects that it is a new message and confirms it is not a retransmission of a message already received,
- Service command code is verified as valid, and
- Service command is permitted for execution in the current RunTime Environment operation mode, by checking the release permission.

Invalid service messages are rejected without further action or response. Errors occurring during the execution of a permitted command are reported by means of an error code in the response to the service message.

The TELEPERM XS side of the TELEPERM XS Gateway computer uses a strictly cyclic processing system similar to that used on the TELEPERM XS computers. The TELEPERM XS side communication protocol is similar to the communication protocol used between all other TELEPERM XS computers and with the Monitoring and Service Interface (MSI). TELEPERM XS Gateway communication is designed with a static memory structure using linear dual-port memory software interface to transmit data values and status information. The non-TELEPERM XS side of the TELEPERM XS Gateway computer reads data according to the external interface independently of the TELEPERM XS side. The standard TELEPERM XS Gateway is designed to be installed on a separate bus from the TELEPERM XS Service Unit. As an alternative, a one-way hardware device can be used to prevent communication from a TELEPERM XS Gateway that is connected to the same bus as the TELEPERM XS Service Unit.

The message transfer protocol used in the design of the TELEPERM XS System ensures that messages sent to the TELEPERM XS computers are secure, valid,

correct, and current. Messages received from the TELEPERM XS Service Unit are checked twice: once by the MSI and again by the TELEPERM XS safety function computers. The standard TELEPERM XS Gateway is not designed to send any messages other than routine signaling messages addressed to the MSI. These design features, when coupled with the lack of software features that are commonly exploited in unauthorized software and data events, eliminate additional software modifications and data security concerns.

These features eliminate the vulnerability to network data storms. The TELEPERM XS message transfer protocol, when coupled with other TELEPERM XS design features, eliminates additional software and data security concerns.

C.4.1.2 Self-monitoring of TELEPERM XS Software Integrity

The integrity of the code installed on the FEPROMs in TELEPERM XS processors is routinely checked for integrity as part of the cyclic test features. This design feature ensures that any damaged code is promptly identified and isolated. Each TELEPERM XS safety function processor performs CRC checks on the FEPROM using the CCITT algorithm for 16 bit cyclic redundancy checksums to ensure the integrity of the code on the FEPROM. This check is performed as a part of the cyclic self test task.

Security-Related Information Withheld
in Accordance with 10 CFR 2.390

C.4.2 TELEPERM XS System Access Controls

The standard TELEPERM XS System is designed to be installed within a secure area. The access control provisions of the TELEPERM XS design conform to IEEE Std 603-1991 and are described in section 2.6 of the TELEPERM XS Topical Report.

Security-Related Information Withheld in Accordance with 10 CFR 2.390

The first layer of control consists of a user login with password check and the assignment of a privilege level to each user account. User accounts and related privilege rights are controlled by plant management. Different privilege levels are assigned to the various tasks and responsibilities of the maintenance personnel, which are granted access to the TELEPERM XS Service Unit. Higher privilege levels are required for personal authorized to access the TELEPERM XS computers. The second layer consists of a key switch release that is required to access any TELEPERM XS computer from the TELEPERM XS Service Unit. The key switch for each TELEPERM XS computer is located in an area under control of the plant operators (e.g., inside the TELEPERM XS equipment cabinets or in the main control room). Activation of the key switch is alarmed in the main control room. The TELEPERM XS mode change alarms enable immediate detection of unauthorized attempts to modify the software.

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Figure C-1: Diverse Means for Access Control to TELEPERM XS Computers

The standard TELEPERM XS access control design features provide reasonable assurance that the software and data security program objectives of confidentiality, integrity, and availability are met. Multiple layers of access control features are provided for the TELEPERM XS Service Unit to prevent unauthorized access to the TELEPERM XS computers. The standard TELEPERM XS Gateway design has redundant logical barriers for access to the TELEPERM XS computers through the use of the MSI and TELEPERM XS Gateway computers.

C.4.2.1 MSI Access Control

As described in the TELEPERM XS Topical Report, the MSI is designed as a qualified data transmission barrier between the TELEPERM XS safety function computers on one side and the TELEPERM XS Service Unit and the TELEPERM XS Gateway on the other side. Not all TELEPERM XS projects will use a TELEPERM XS Gateway.

The MSI is designed with the same TELEPERM XS operating principles as the TELEPERM XS safety function computers, as described in Section C.3. The MSI does not function as part of the protection channels. The MSI is designed to only check configured communication channels that are defined during code generation. These design features ensure that unexpected or improper messages are ignored. All software executed on the MSI is safety-related, thus is developed and tested accordingly.

The MSI is the only physical network connection that exists between the TELEPERM XS safety function processors and non-safety systems (e.g., TELEPERM XS Gateway or TELEPERM XS Service Unit). The TELEPERM XS Gateway is also isolated from the TELEPERM XS Service Unit by a MSI or the use of a one-way hardware device. No TCP/IP or message routing services are installed on the MSI. These design features eliminate important transmission vectors for unauthorized external software modifications and data events.

The MSI also acts as a qualified failure propagation barrier for the communication between the (non-safety) TELEPERM XS Service Unit or TELEPERM XS Gateway and the TELEPERM XS computers. TELEPERM XS safety processor computers are not directly accessed by the TELEPERM XS Service Unit; instead, they can only be accessed via the MSI. A failure in the communication link between the MSI and TELEPERM XS Service Unit or the TELEPERM XS Gateway removes any possibility of these devices from accessing the TELEPERM XS safety processors. The effect of these failures does not compromise the inherent software and data security features of

the MSI or the TELEPERM XS safety processors. The TELEPERM XS System is designed such that the loss of the MSI computer does not lead to a degradation of the safety-related protection functions.

The MSI is designed to check for and only use information from expected data, signaling and service messages, which are defined during code generation. Pre-defined data messages are generated from safety function processors and sent to the TELEPERM XS Gateway via the MSI. The MSI is designed to:

Security-Related Information Withheld in Accordance with 10 CFR 2.390

The TELEPERM XS Service Unit does not process TELEPERM XS Application Software, therefore it is not designed to receive or send data messages. Consequently, it is not possible to engineer data messages between the Application Software of some TELEPERM XS computers or the MSI and the TELEPERM XS Service Unit.

The MSI is designed to check for and only use data from expected messages, which are defined during code generation. The MSI is also designed to only check configured communication channels. These design features ensure that unexpected or improper messages are ignored. Pre-defined signaling and service messages utilizing proprietary message structure and the protected media access control (MAC) address

Security-Related Information Withheld in Accordance with 10 CFR 2.390

of the TELEPERM XS Service Unit or the TELEPERM XS Gateway mitigate the threat of unauthorized traffic.

C.4.2.2 TELEPERM XS Service Unit Access Control

The TELEPERM XS Service Unit is used by authorized personnel to monitor and test the TELEPERM XS System, to diagnose system alarms and failures, and to make parameter and software changes. The TELEPERM XS safety function computers are not directly accessed by the TELEPERM XS Service Unit; instead the TELEPERM XS safety function computers are only accessed via the MSI, which provides communication independence. The TELEPERM XS System and the associated TELEPERM XS Service Unit are located in secure areas. The identification of the TELEPERM XS Service Unit by the MSI is accomplished by a protected MAC Address used to decode incoming service messages.

Additional information on the TELEPERM XS Service Unit interface is provided in Annex 1 to this Appendix.

C.4.2.3 TELEPERM XS Gateway Access Control

The standard TELEPERM XS Gateway is designed to support two-way communication via the MSI to provide communication independence. The standard TELEPERM XS Gateway is designed to primarily export TELEPERM XS System information to the plant process computer or other non-TELEPERM XS systems. The standard TELEPERM XS Gateway is also designed to send routine signaling messages and a limited set of data messages back to the MSI. The MSI 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 software and data security controls. Not all TELEPERM XS projects will use a TELEPERM XS Gateway.

C.4.2.4 TELEPERM XS Safety Processor Access Control

The TXS equipment cabinets are normally locked. TXS System equipment physical access controls include the following:

Security-Related Information Withheld in Accordance with 10 CFR 2.390

The TELEPERM XS cabinet doors alarms enable immediate detection of unauthorized access.

C.4.2.5 Maintenance Laptop and Test Machine Access Control

The primary use of the maintenance laptop is to perform initial software load on a new processor board (e.g., after maintenance repair). It is used for upgrades to the System Segment Software and to configure L2-CP communication modules. The maintenance laptop can also be used to load Application Software revisions on processor boards (e.g., during an outage or if the Service Unit is not available). It can also be used to retrieve diagnostic failure information from exception handler buffer to diagnose failures; however, this use is not a typical user maintenance activity but may be used during commissioning testing.

The maintenance laptop connects to serial interface X4.1 on the front of a SVEx processor using a special boot dongle connector. The SVEx processor can only load software from the X4.1 interface with the processor in boot load mode (i.e., not in normal operation). The SVEx processor can only perform the diagnostic functions from the X4.1 interface with the processor in diagnostic monitor mode (i.e., not in normal operation). The maintenance laptop uses the **sveload** tool to load the software modules and uses the Debug Spooler Task software to perform the diagnostic activities. The use of the **sveload** software requires a special license dongle.

The same verification methods described in Annex 2 Section 6.8 are used for software loaded from the maintenance laptop or Service Unit and are performed from the Service Unit.

The X4.1 interface can only be accessed by opening the TELEPERM XS cabinet door, which generates a control room alarm. The reset of a TELEPERM XS processor (to enter boot load or diagnostic monitor modes) also generates a control room alarm. The serial cable must be removed from X4.1 interface after use. The serial cable cannot extend outside of security zone.

The maintenance laptop cannot access TELEPERM XS RunTime Environment to change modes. It cannot be used to modify changeable parameters. It cannot be used to initiate signal tracing or issue service requests. The predefined communication channels in TELEPERM XS System cannot be changed via the maintenance laptop. The maintenance laptop cannot access other local software when in Diagnostic Monitor Mode.

A connection is provided for a dedicated maintenance test machine to be connected to the TELEPERM XS Service Unit to perform maintenance or test activities. The maintenance test machine connects to Service Unit and TELEPERM XS System input/output (I/O) devices; it does not connect directly to the SVEx modules. The maintenance test machine is only connected when its use is required.

The maintenance laptop (including the X4.1 interface connection cable) and test machine are controlled in accordance with the requirements of the plant software and data security plan required by 10 CFR 73.54. Typically controls for any laptop or test machine connection to the TELEPERM XS System may include the following:

Security-Related Information Withheld in Accordance with 10 CFR 2.390

The administrative controls in place for the maintenance laptop and test machine provide software and data security protection from unauthorized activities attempting to introduce or utilize unrecognized software vulnerabilities.

C.4.2.6 Remote Access Control

The TELEPERM XS System does not have remote access capability. This system feature provides software and data security protection from unauthorized activities attempting to exploit unrecognized software vulnerabilities.

C.4.3 TELEPERM XS System Software and Data Security Features Summary

The TELEPERM XS System design employs several layers of physical access controls. These features were designed into the system to meet the control of access requirements specified in Section 5.9 of IEEE Std 603-1991. The layers of access control features provided in the TELEPERM XS System design provide the software and data security protection for unauthorized activities from inside the plant that attempt to insert or utilize unrecognized software vulnerabilities.

The TELEPERM XS Gateway access controls and the limitations on remote access capabilities provide the software and data security protection for unauthorized activities from outside the plant that attempt to insert or utilize unrecognized software vulnerabilities.

The MSI is designed as a qualified data transmission barrier to the inner security zone (i.e., the safety function processors) in the TELEPERM XS System. The MSI provides an additional barrier to certain software and data threats from the TELEPERM XS Gateway or Service Unit. The system is designed such that the loss of the MSI computer (e.g., as a result of an attempted unauthorized software and data event) does not lead to a degradation of the safety-related protection functions.

C.5 TELEPERM XS Software Development Process Controls

TELEPERM XS software consists of TELEPERM XS System Software code (produced by AREVA NP GmbH) and TELEPERM XS Application Software code (produced by AREVA NP Inc.). All TELEPERM XS code is fully documented to the requirements specified for the software integrity level of the code. No third party safety software is used in the TELEPERM XS System.

The procedure controls used for the development of TELEPERM XS software described below provide a secure software development infrastructure.

C.5.1 Process Controls for Software Developed and Maintained by AREVA NP GmbH

The standard TELEPERM XS System is described in the TELEPERM XS Topical Report, which describes both the design features and the development process for the TELEPERM XS hardware platform and System Software. The software and data security controls for the development of the TELEPERM XS System Software and Function Block library software are briefly discussed in TELEPERM XS Topical Report Section 3.2.1.2, Qualification Process. AREVA NP GmbH procedure FAW 1.7 (Reference 68) covered security aspects during the development.

The procedure controls in effect during the original design of the TELEPERM XS System Software and Function Block library included the following requirements that provided a secure software development infrastructure:

Security-Related Information Withheld in Accordance with 10 CFR 2.390

The information security requirements have expanded since the TELEPERM XS Topical Report was issued. The information security requirements in place at that time were described in Engineering Procedure FAW-1.7 was applicable to TELEPERM XS

activities. The information security requirements for AREVA NP GmbH have been expanded to encompass all business activities, including TELEPERM XS. An AREVA NP GmbH corporate-wide procedure on information technology security incorporated all of the requirements from FAW 1.7 requirements and superseded that procedure.

The procedure controls currently in effect for the TELEPERM XS System Software and Function Block library include the following requirements that provide a secure software development infrastructure:

Security-Related Information Withheld in Accordance with 10 CFR 2.390

NRC conducted an inspection of the TELEPERM XS information security program in Erlangen, Germany on March 10-14, 2008. The following excerpt is taken from NRC

Inspection Report for AREVA NP GmbH 99901371/2008-201, dated May 7, 2008 (Reference 47):

> The inspectors requested engineering procedure FAW-TELEPERM XS 1.7, "Information Security," dated August 23, 2004, to review the security aspects during the development of the TELEPERM XS operating system software and function block library software to determine whether AREVA has a secure software development infrastructure. The vendor indicated that a corporate-wide procedure for information technology security incorporating all of the FAW 1.7 requirements had superseded this procedure. The inspectors reviewed the new procedure, AREVA NP GmbH "Informationssicherheit," dated November 25, 2005, with support from the vendor's technical staff. The inspectors also conducted detailed interviews with AREVA personnel to ensure that this new replacement procedure covered the necessary security aspects and requirements of a secure software development infrastructure. The inspectors requested information regarding, but not limited to, isolation of the development computers/network from the corporate network, access control and access rights to particular development computers, process for using purchased software, and physical access restrictions to the development lab. Additionally, the inspectors verified implementation of security measures during the tour of the TELEPERM XS platform development lab. The inspectors found no issues associated with the security controls used for the development of TELEPERM XS software.

The AREVA NP GmbH information technology access controls applied to desktop computers and laptops used in the AREVA NP GmbH corporate network. Standard AREVA NP GmbH policy does not allow remote access to the development networks.

Security-Related Information Withheld
in Accordance with 10 CFR 2.390

from all other computers interlinked in the AREVA-internal network. Access to the development network via this firewall is exclusively granted to working place computers and accounts of the development department employees. When working on the TELEPERM XS design, the standard working place computers of those employees are operated as smart clients logged on the development server computers of the protected network. Source code and builds are therefore kept locally in the protected network when performing development work. Intrusion detection is effective on all computers.

Password diversity, changing passwords from their default settings, and the use of strong passwords are standard AREVA NP GmbH policy. Passwords are required to be changed every three months as standard AREVA NP GmbH Policy. System lock out after several failed logins is the standard AREVA NP GmbH policy. The development network system administrator handled resets and unlocking of passwords in accordance with standard AREVA NP GmbH Policy.

C.5.2 Process Controls for Software Developed and Maintained by AREVA NP Inc.

The development process for TELEPERM XS Application Software for U.S. projects is described in this Topical Report. The procedure controls currently in effect for the TELEPERM XS Application Software for U.S. projects include the following requirements that provide a secure software development infrastructure:

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Access to the Engineering Server is maintained through the Information Technology (IT) servers which utilize corporate firewalls and virus protection in accordance with established policies and procedures. The network structure has the following characteristics:

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

System administrators have full access to the Application Software development environment. Ordinary users (i.e., employees with access to the development building and development network whom are involved in the development of TELEPERM XS software) are prohibited from modifying project files or directories without the consent of the system administrators. Read-only access is granted unless the access rights are modified by the system administrators. Standard AREVA NP Inc. policy does not allow remote access to the development server.

Modifications by users to their workstation are limited to the user profile. Installations of software tools used for the development of TELEPERM XS software and software configuration management are under administration of the system administrator and commonly used by the team of employees. Unauthorized modifications to those

installations or the data repository could therefore become detected by some other employee of the team.

The AREVA NP Inc. corporate IT department constantly scans networked computers for illegal software and any identified computers with illegal software will be reformatted.

The AREVA NP Inc. software library is administratively controlled by a software librarian. The controlling procedure 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 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.

C.5.3 Undocumented Software Protection

The TELEPERM XS System Software development process has several layers of protection to ensure TELEPERM XS System Software does not contain undocumented code, and other unauthorized and undocumented functions or applications. The main protective measures employed for TELEPERM XS System Software are:

- Restricted access to the source code,
- Source code reviews,
- Automated build procedures,

- Tool based module tests with test coverage analysis,
- Component tests,
- Integration tests,
- Identification by checksums, and
- Qualification (third party assessment) of the software platform.

Each software development step is followed by an internal review. Development document reviews and source code reviews are performed by both, peer developers and independent assessors.

The TELEPERM XS Application Software development process has several layers of protection to ensure TELEPERM XS System Software does not contain undocumented code, and other unauthorized and undocumented functions or applications. The main protective measures are:

- Restricted access to the TELEPERM XS Application Code development tools and code files,
- Automated code generation (using only a qualified and verified version of the SPACE Engineering Tool and Function Block Library),
- Independent validation testing, and
- Software file identification by checksums.

Each software development step is followed by an internal design review. Development document reviews are performed by the independent Verification and Validation (V&V) group, which also performs the validation testing.

Annex 2 to Appendix C provides a full description of the process controls that provide undocumented software protection for the TELEPERM XS Software development processes.

C.6 Alignment with Regulatory Guide 1.152 Software and Data Security Guidance

The generic TELEPERM XS System design considered security issues from the earliest stages. The key design features related to current software and data security are described in TELEPERM XS Topical Report Sections 2.6 (Control of System Access), 2.7 (Fault Tolerance Features), and 2.9 (Interference Free Communication). The security aspects of the generic TELEPERM XS platform development infrastructure were described briefly in TELEPERM XS Topical Report Section 3.2.1.2 (Software Development – Qualification Process). The engineering procedure FAW-1.7 was submitted to NRC in a September 1, 1999 letter supporting the TELEPERM XS Topical Report review (Reference 48).

C.6.1 Concepts Phase

The key security concepts considered for the TELEPERM XS System included:

- Compliance with IEEE Std 603-1991 control of access requirements (clause 5.9), which permits the use of administrative control of access to safety system equipment. The standard requires that the administrative controls shall be supported by provisions within the safety systems, by provision in the generating station design, or by a combination thereof.
- Faults in the non-safety related portions of the TELEPERM XS System (i.e., TELEPERM XS Service Unit and Gateway) must not affect the operation of the safety-related portions of the system.
- Location of TELEPERM XS equipment within a secure area.
- Development of safety-related software in a controlled environment.

A security assessment, as described in Regulatory Guide 1.152 was not performed for the generic TELEPERM XS System design since that regulatory guidance was not in effect at the time of the system development. However, software and data security

issues were considered in the TELEPERM XS System development, as documented in the TELEPERM XS Topical Report.

A typical diverse and defense-in-depth analysis (References 48, 50, 56, and 57) was performed for the TELEPERM XS System demonstrated that plant safety can be ensured even if some postulated unauthorized software and data event was successful at exploiting some unrecognized software vulnerability and preventing the TELEPERM XS System from performing its safety function.

Remote access capability was not implemented for the TELEPERM XS System, as noted in Section C.4.2.6.

The implementation of these requirements was later influenced by the evolving software and data security guidance, as noted by Regulatory Guide 1.152, NEI 04-04 (Reference 42), and DI&C-ISG-01 (Reference 19).

The concept phase of a TELEPERM XS project addresses the preliminary design concepts which address software and data security: access control, key switches, control room alarms, TELEPERM XS equipment location, and TELEPERM XS Gateway interfaces (if applicable).

C.6.2 Requirements Phase

The generic TELEPERM XS System development process considered generic and project-specific software and data security aspects for the TELEPERM XS System during the requirements phase. The standard TELEPERM XS System platform (hardware and System Software) was designed several years prior to the issuance of Revision 2 of Regulatory Guide 1.152. As such, some elements of the generic system qualification are not explicitly addressed as software and data security activities in the TELEPERM XS Topical Report and the associated NRC Safety Evaluation Report. However, it must also be recognized that aspects of the TELEPERM XS platform design that address the nuclear safety aspects of communication independence, safety to non-

safety system isolation, interference-free communication, and access control are equally applicable to the software and data security attributes of confidentiality, integrity, and availability.

From the very beginning of the TELEPERM XS System platform development AREVA NP (formerly: Siemens KWU) decided to design a system meeting the highest safety requirements. The hardware modules were selected to a high degree from pre-existing equipment families while the modules of the system software and the SPACE tool were developed following the stipulations of IEC 60880 (Reference 43). All platform features important to safety were tested, assessed and certified by independent experts Gesellschaft für 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).

In order to ensure highest quality for the Application Software engineering as well, specific emphasis was given to the development of the SPACE Engineering Tool automatic code generators, as well as of tools which effectively support the verification and validation of the Application Software.

AREVA NP Inc. does not use commercial off the shelf software in its TELEPERM XS safety system. All TELEPERM XS System Software and development tools are purchased and received from AREVA NP GmbH. Each software package is developed utilizing the same processes described in the TELEPERM XS Topical Report.

The requirements phase of a TELEPERM XS project addresses the project-specific security requirement for access control, key switches, control room alarms, TELEPERM XS equipment location, and TELEPERM XS Gateway interfaces. The independent V&V group performs a security assessment activity that ensures the correctness, completeness, accuracy, testability, and consistency of the system security requirements.

Annex 2 Sections 4 and 6 describe the layers of protection in the requirements phase of the development process to ensure TELEPERM XS Software does not contain undocumented code, and other unauthorized and undocumented functions or applications.

C.6.3 Design Phase

The TELEPERM XS System was designed with a number of features that minimize or eliminate software and data security vulnerabilities, as described in Section C.4. The TELEPERM XS System design uses a combination of personnel access controls, administrative controls, and design features (including hardware and software elements with appropriate alarms) to protect the safety-related software from unauthorized alteration. The design features are a combination of TELEPERM XS platform/system design features (e.g., self surveillance and CRC tests) and application design features (e.g., physical key switches for control of safety function processor modes). These features provide software and data security protection from unauthorized activities (i.e., from inside or outside the plant) attempting to utilize unrecognized software vulnerabilities.

The design phase of a TELEPERM XS project addresses the specific design features which address the project-specific software and data security requirements for access control, key switches, control room alarms, TELEPERM XS equipment location, and TELEPERM XS Gateway interfaces. The independent V&V group verifies that the security requirements have been effectively implemented in the design.

Annex 2 Sections 4 and 6 describe the layers of protection in the design phase of the development process to ensure TELEPERM XS Software does not contain undocumented code, and other unauthorized and undocumented functions or applications.

C.6.4 Implementation Phase

Highly deterministic behavior of the TELEPERM XS System was a key development target for the TELEPERM XS platform. Another information element in the development of the TELEPERM XS System was a high quality development process. The main components of the TELEPERM XS System platform are:

- the hardware modules,
- the system software modules,
- the set of software engineering tools,
- the engineering instructions, and
- the standard platform documentation together with certificates on safety features and test reports.

The details of the TELEPERM XS System and the development process were described in the TELEPERM XS Topical Report.

The development process for the qualified components of the TELEPERM XS System platform included comprehensive V&V activities by the manufacturer, as well as by an external appraisal of the development and test results by test institutes (GRS, ISTec, and TÜV). The V&V activities which are performed during development comprise:

- Reviews of the development documentation and test specifications
- Module tests of the vital software components – All possible branches within the function to be tested are executed in this process. The major part of the functionality of the software can be verified with this test. For software modifications, testing included a regression test and a test of the implemented modifications.
- Code inspections - A code inspection of the complete source code was performed. For minor software modifications, the code inspection may be limited to the modified sections of the code.

- Functional tests of the software components - A component test was performed for all modified software components.

The modified software modules were subjected to an external qualification test by GRS, ISTec, and/or TÜV. The type test program was based on the following standards:

- KTA 3503, Type Testing of Electrical Modules of the Reactor Protection System, November 1986 edition (Reference 44)
- IEC 60880-1986, Software for Safety Systems in Nuclear Power Stations

A qualification certificate was issued by the test institute for each qualified software module demonstrating that the module was qualified by test to the specified requirements.

All TELEPERM XS software is developed and documented by AREVA NP and no commercial off the shelf programs are used in its TELEPERM XS safety system.

The use of the SPACE Engineering Tool to develop the project-specific Application Software is mandatory. The SPACE Engineering Tool Code Generator tools have been designed and qualified for the creation of safety application software (ANSI C source code). They include a variety of checks of the specification data and follow strict design rules for the generated code meeting IEC 60880 stipulations. The project specific Application Software has to be generated entirely by means of these tools in order to ensure that all design rules and interfaces defined for TELEPERM XS Application Software functions are met. The use of the SPACE Engineering Tool eliminates the direct human interface with the application software source code.

The implementation phase of a TELEPERM XS project addresses the specific activities which translate the project-specific software and data security design requirements for access control, key switches, control room alarms, TELEPERM XS equipment location, and TELEPERM XS Gateway interfaces into the project-specific hardware and Application Software. The independent Verification and Validation group verifies that

the security design requirements have been effectively implemented in the project-specific hardware and Application Software.

Annex 2 Sections 4 and 6 describe the layers of protection in the implementation phase of the development process to ensure TELEPERM XS Software does not contain undocumented code, and other unauthorized and undocumented functions or applications.

C.6.5 Test Phase

Integration and System Test were performed on the generic TELEPERM XS System by ISTec and TÜV as part of the independent qualification process.

The following system characteristics relevant to the software and data security features were confirmed during the external qualification testing:

- Processing and communication cycle times are not influenced by external process states (measured signals, amount of alarms and monitored information).
- Mutually independent I&C functions are processed as specified according to their chronological order and their input signals.
- Mutually independent processing units do not affect each other regarding their operating modes and their time behavior. Processing units which exchange signals but are otherwise mutually independent have only effect on each other's time response within the limits of the engineered communication functions.
- Interference on cables with violation of the measuring range and input module failures are detected, marked as signal failures, and indicated. Signals detected as faulty are processed and indicated by the system components (RunTime Environment, I/O drivers, and function blocks) as defined in the specification.
- Transmission failures on TELEPERM XS Ethernet (H1) and TELEPERM XS Profibus (L2) busses are detected, processed, and indicated in accordance with the specification. Single message failures are tolerated by the system. Furthermore, on TELEPERM XS Ethernet (H1) busses double message failures

are tolerated. Interference caused by a receiving unit on the sending one is impossible.

- Sending and receiving processing units execute their functions asynchronously if no "expedited messages" are sent via serial bus links, with the exception of voter subunits monitoring each other. Lost messages are treated like transmission errors. Thus failures of individual sending processing units are always tolerated if signal information is distributed via redundant trains and special fault propagation barrier function blocks are used on the receiving processing modules.
- Single failures of active and passive hardware modules are detected and indicated corresponding to the implemented monitoring mechanisms (self-monitoring, monitoring of the communication, cabinet annunciation system). Multiple failures are detected and indicated if sufficient resources (for example communicating processing units and communication processors) are provided. The cabinet annunciation system is activated according to the specification.
- Fault propagation barriers are effective provided that no plant-specific fault suppression measures are engineered (for example status correction). Signal status is changed by the RunTime Environment as specified, i.e., if required, status is changed to ERROR but never from ERROR or TEST to OK.
- The RunTime Environment behaves in the operating modes start-up, operation, parameterization, functional test, and diagnosis as specified. It changes between operating modes according to the specification. Permissive signals for operating modes are designed individually according to project requirements and are not dealt with in the integration test.
- The RunTime Environment can be controlled by means of service requests. Disabling and enabling of service requests are effective as required for the respective operating mode.
- The user software can be loaded from a centralized unit using the network connections. This function can be deactivated by a hardware switch on the processing modules.

- Fail-safe behavior: Signals marked as faulty (ERROR and/or TEST status) are issued as 0 signals via output modules. Exceptions cause output of 0 signals via output modules and cause shut down or restart of the computers affected.

On completion of the external qualification test, a test certificate was issued by the independent expert for each qualified software component and a test report was written. The test certificate documents that the TELEPERM XS System met the criteria listed above.

AREVA NP met with the NRC in Erlangen, Germany on December 8, 2008 and again on April 15, 2009 in Alpharetta, Georgia regarding the design of the TELEPERM XS MSI (see References 51 and 52). The meetings were held to address the NRC's questions about generic TELEPERM XS platform testing that addresses certain issues of concern: immunity from software and data threats via the TELEPERM XS Gateway and immunity from spurious signals from the TELEPERM XS Service Unit. AREVA NP made a presentation describing how the MSI acts as a data transmission barrier to the inner security zone in TELEPERM XS Systems. As such it provides a barrier to certain threats from the TELEPERM XS Gateway and TELEPERM XS Service Unit. The purpose of the presentation was to explain recent fault injection testing performed by AREVA NP that supplements the original MSI qualification testing that addresses NRC's areas of concern.

The presentation was provided in two parts: the TELEPERM XS Gateway to MSI interface and the TELEPERM XS Service Unit to MSI interface. The first presentation addressed the following main points:

- IT Security Zone Model
- Requirements to be met by the MSI
- Test Concept
- Test Platform: Hardware and Software
- Test Cases and Test Procedure

- Test Results

The test report concludes that in the case of bidirectional exchange of data messages between the TELEPERM XS Gateway and MSI, message interchange within the inner security zone cannot be affected by manipulation on the TELEPERM XS Gateways or communication links of the outer security zone (i.e., no generation of additional messages or disabling of engineered messages). Data messages can only transfer engineered signals (i.e., no modification of underlying data structure is possible). This report shows that data communication in TELEPERM XS System meets the security requirements of IEEE Std 603-1991 and IEEE Std 7.4.3.2-2003, as well as those of NRC Regulatory Guide 1.152.

The second presentation addressed the following main points:

- IT Security Zone Model
- Requirements to be met by the Service Unit
- Requirements to be met by the MSI
- Requirements to be met by the Online System
- Test Concept
- Test Platform: Hardware and Software
- Test Cases and Test Procedure
- Test Results

The testing demonstrates that the MSI acts as an effective barrier to filter spurious messaging from the TELEPERM XS Service Unit to satisfy IEEE Std 603-1991 requirements for the safety to non-safety interface. The testing also demonstrates that the MSI in conjunction with the other TELEPERM XS access control features (i.e., mode change release requirements and alarms) ensures that the TELEPERM XS System meets the security requirements of IEEE Std 603-1991 and IEEE Std 7.4.3.2-2003, as well as those of NRC Regulatory Guide 1.152 for the TELEPERM XS Service Unit interface.

The MSI testing was evaluated by the independent test institutes: ISTec and TÜV Nord. The following test documents were discussed:

Original MSI Barrier Qualification

- ISTec and TÜV Nord Certificate No. TELEPERM XS-MSI-1102-01
- ISTec Test Report for Certificate No. TELEPERM XS-MSI-1102-01

Recent MSI Testing

- ISTec Certificate No. TELEPERM XS-MSI-0308-03
- ISTec Test Report for Certificate No. TELEPERM XS-MSI-0308-03

The test phase of a TELEPERM XS project validates the specific system and software design requirements for access control, key switches, control room alarms, TELEPERM XS equipment location, and TELEPERM XS Gateway interfaces were properly implemented in the TELEPERM XS System.

The project-specific TELEPERM XS system and acceptance tests are designed to validate the correct functionality of the integrated system (i.e., with all software implemented, with all interfaces and all peripheral equipment that is in the scope of the delivery). This integrated system includes:

- User Interfaces
- Hardware Interfaces
- Software Interfaces
- Communications Interfaces

The testing methodology ensures that the software and data security requirements and features have been designed and implemented correctly, including but not limited to external hardware connections (e.g., keys switches and alarms) and external communication pathways (e.g., TELEPERM XS Gateway).

Annex 2 Sections 4 and 6 describe the layers of protection in the test phase of the development process to ensure TELEPERM XS Software does not contain undocumented code, and other unauthorized and undocumented functions or applications.

C.7 Summary

The scope of the AREVA NP software and data security responsibilities include the Concept Phase through the Test Phase, as described in Regulatory Guide 1.152. The combination of the information in the TELEPERM XS Topical Report on system design features, the information in the TELEPERM XS Software Program Manual on Application Software development, and the specific information provided in this document fully address the software and data security issues associated with the TELEPERM XS System.

Applicants using the TELEPERM XS technology for safety-related projects will need to address the software and data security items for Installation, Checkout and Acceptance Testing, Operation, Maintenance, and Retirement Phases. The applicant must address software and data security requirements for Gateway communication outside the most secure defensive layer.

Annex 1 - TELEPERM XS Service Unit Interface

1.0 Background

The design and use of the TELEPERM XS Service Unit was extensively described in the TELEPERM XS Topical Report. The following sections from the report describe relevant aspects of the Service Unit, its use, and its isolation from the safety-related portions of the system:

2.4.3.4.1	Operating System
2.4.3.4.4	Function Block Modules
2.5.1	Introduction (to Testability)
2.5.2	Tasks of the Service Unit
2.5.3.4	Online Display of Hardware Structure
2.5.4	Connection to the I&C System
2.5.5	Architecture of the Service Unit
2.5.6	Handling and Use of the Service Unit
2.5.7	Testing and Maintenance
2.6	Control of System Access (in particular, 2.6.2 Technical Measures)
2.9	Interference-Free Communication (in particular, 2.9.3 Ethernet Communication)
3.1.1.1	System Overview
3.1.1.4	Requirements and Architectural Challenges
3.1.3.4	RunTime Environment
3.2.2.2	Goals of the Integration and System Tests
3.2.2.3	Qualification Process
3.2.2.4	Submitted Documents and Results
4.1	Typical Architecture of Safety I&C Systems
4.2	Design Principles
4.3	Signal Transmission from Class 1E to non 1E Equipment and Circuits
4.4	Single Wire Signal Transmission
4.5	Digital Data Transmission via Serial Busses (in particular, 4.5.1 Communication with the Service Unit)
7.6	Independence
7.9	Control of Access
7.20	Maintenance Bypass
8.1.1	Overall System (in Qualification Documents)

The NRC Safety Evaluation Report for the TELEPERM XS Topical Report discusses the TELEPERM XS Service Unit, its use, and its isolation from the safety-related portions of the system:

Section 2.0	SYSTEM DESCRIPTION (at pages 5 and 6)
Section 2.2	Software Description (at pages 15 and 16)
Section 2.2.1.1	Operating System Software (at pages 19 and 20)
Section 2.2.1.2	Platform Software (at pages 21, 22, and 27)
Section 5.0	SUMMARY OF REGULATORY COMPLIANCE EVALUATION (at pages 47, 51, and 52)

2.0 NRC Guidance Applicable to TELEPERM XS Service Unit Interface

NRC document DI&C-ISG-04 (Reference 20) states in the IMPLEMENTATION section that:

> "Except in those cases in which a licensee proposes or has previously established an acceptable alternative method for complying with specified portions of the NRC's regulations, the NRC staff will use the methods described in this Interim Staff Guidance (ISG) to evaluate licensee compliance with NRC requirements as presented in submittals in connection with applications for standard plant design certifications and combined licenses."

The relevant aspects of the TELEPERM XS Service Unit design, its use, and its isolation from the safety-related portions of the system were extensively described in the TELEPERM XS Topical Report and approved by the NRC in the associated Safety Evaluation Report.

Criterion 10 in DI&C-ISG-04 Section 1, Interdivisional Communications, is directly related to the TELEPERM XS Service Unit design. It states:

> 10. Safety division software should be protected from alteration while the safety division is in operation. On-line changes to safety-related system software should be prevented by hardwired interlocks or by physical disconnection of maintenance and monitoring equipment. A workstation (e.g. engineer or programmer station) may alter addressable constants, setpoints, parameters, and other settings associated with a safety function only by way of

> the dual-processor / shared-memory scheme described in this guidance, or when the associated channel is inoperable. Such a workstation should be physically restricted from making changes in more than one division at a time. The restriction should be by means of physical cable disconnect, or by means of keylock switch that either physically opens the data transmission circuit or interrupts the connection by means of hardwired logic. "Hardwired logic" as used here refers to circuitry that physically interrupts the flow of information, such as an electronic AND gate circuit (that does not use software or firmware) with one input controlled by the hardware switch and the other connected to the information source: the information appears at the output of the gate only when the switch is in a position that applies a "TRUE" or "1" at the input to which it is connected. Provisions that rely on software to effect the disconnection are not acceptable. It is noted that software may be used in the safety system or in the workstation to accommodate the effects of the open circuit or for status logging or other purposes.

The AREVA NP assessment of the TELEPERM XS System alignment with DI&C-ISG-04 noted the following alternative:

> The TELEPERM XS System uses a combination of hardware and software features to control access to the TELEPERM XS safety-processors. The mechanical key switches do not specifically interrupt the signal path but do so via software. A safety processor receiving a Service Unit command to change a parameter must also have a Change Enable command input via keyswitch before it will consider the command to be valid. The software access controls reside within the safety-processor. Access is provided via the Service Unit, which connects to the safety-processors through the MSI. The Service Unit is always connected to the MSI to provide monitoring and to support constant busload operation. Administrative controls are used to limit access to one division at a time during plant operation.

The TELEPERM XS System design has a previously established acceptable alternative method for complying with this criterion, as documented in the NRC Safety Evaluation Report for the TELEPERM XS Topical Report.

3.0 First-of-a-Kind Implementation Audit

NRC conducted an audit of the TELEPERM XS Service Unit interface design for the Oconee Reactor Protection System/ Engineered Safeguard Protection System design at

the AREVA NP facility in Alpharetta, Georgia on April 15 – 17, 2009 (Reference 53). The audit was based on the presentation information submitted to NRC in Reference 52.

The failure modes and effects for the failures affecting the Parameter Change Enable Application Software logic were assessed for implications on control of undetected access to the safety processors or undetected changes of processor modes. The following conditions were evaluated:

- Failure of the Parameter Change Enable key switch
- S430 Module Failure
- Random bit error in random access memory (RAM) for Changeable Parameters affecting safety functions
- Random bit error in RAM for Changeable Parameters affecting Channel E functions
- Random bit error in RAM affecting RunTime Environment
- Stuck bit in RAM
- Bit error in FEPROM

No single failure affecting the TELEPERM XS Parameter Enable Logic allows undetected access to the safety processors or undetected changes of processor modes.

The design of the Service Unit provides important system monitoring capabilities. Disconnection of the Service Unit adversely affects system monitoring capabilities, such as data archival and error logging that are essential for trouble-shooting activities and software and data security access tracking. Elimination of the Service Unit continuous monitoring function would eliminate data archival and error logging information that is essential to safe and secure plant operation. Delays in troubleshooting activities would have an adverse impact by increasing system unavailability, which is contrary to the NRC's Maintenance Rule (Reference 2). Elimination of log information will hamper

software and data security logging and monitoring capabilities, which is contrary to the 10 CFR 73.54.

The NRC audit report made the following statements:

2.3 Data Communications Objectives

The objective for the review of the communications methods was to understand the data methods, including operation of the MSI, to determine whether failure of communications would compromise the independence of the various divisions and prevent them from performing the safety function in a reliable and deterministic manner. In order to achieve this, the NRC staff reviewed the following:

- Software involved with the acceptance of messages from outside of each division to determine how inappropriate messages for the current operating mode are rejected. This review required examination of the actual lines of code contained within the function blocks to determine how the filtering is accomplished and how the filtering changes when the key switch is turned in a different position.
- Operation of the MSI, including what modules were being used, how those modules were connected, and the signal flow through modules and MSI. The connectivity of the MSI and the service unit was reviewed.
- Operation of SL21 and SCP2 communication modules. The review included the type and operation of the communications microprocessor and operation of the dual-ported memory used to transfer information from the communications modules to the safety processer (sic).

The review was facilitated by presentations by AREVA GmbH staff. The following are the presentations:

- Requirements and Test of the Service Unit/Online Systems Barrier
- TELEPERM XS Permissions and Operating Modes
- TELEPERM XS RunTime Environment Parameter Change Enable
- Testing Functionality of Parameter Change Enable Keyswitch

2.4 Communications Data Summary

The NRC staff reached a sufficient understanding of the data communications methods as proposed for use at Oconee to conclude that

guidance of the integrated scheduling group (sic) #4 are either being met or an acceptable alternative method was being used. The NRC staff expects that any further questions concerning data communications will be confirmatory and not to reach a determination of compliance with regulatory requirements.

4.0 Summary

In summary, the TELEPERM XS Service Unit design is based on an approved TELEPERM XS System design and augmented with appropriate project-specific features to provide the necessary software and data security protection. The combination of the system design requirements and administrative controls of the keyswitches provide equivalent or greater assurance against susceptibility to failure or unauthorized intrusion than the conservative alternative described in DI&C-ISG-04 Section 1 Criterion 10. The TELEPERM XS design features related to the Service Unit interface were validated by the NRC during a first-of-a-kind implementation audit.

Annex 2 – TELEPERM XS Software Development (Code Protection)

1.0 Introduction

Security-Related Information Withheld in Accordance with 10 CFR 2.390

2.0 Scope

The TELEPERM XS System Software development specific topics include an analysis of unauthorized activities by a user or developer during all development phases. TELEPERM XS System Software consists of the generic System Software employed on the computers of a TELEPERM XS I&C System. In the context of this report the TELEPERM XS System Software includes all relevant TELEPERM XS Application Software development tools (i.e., code generators, compiler, linker, and locater) for building the Application software. Main protective measures employed for TELEPERM XS System Software are:

- Restricted access to the source code,

- Source code reviews,
- Automated build processes,
- Tool based module tests with test coverage analysis,
- Component tests,
- Integration tests,
- Identification by checksums, and
- Qualification (third party assessment) of the software platform.

Each software development step is followed by an internal review. Development document reviews and source code reviews are performed by both, peer developers and independent assessors. Sections 3.0 and 4.0 of this Annex are applicable to the TELEPERM XS System Software development work performed by AREVA NP GmbH in Erlangen, Germany.

The TELEPERM XS Application Software development specific topics include an analysis of unauthorized activities by a user or developer during all development phases. TELEPERM XS Application Software consists of the TELEPERM XS network and logic design performing the safety related logic employed on the computers of a TELEPERM XS I&C system running TELEPERM XS System Software. In the context of this report, the TELEPERM XS Application Software includes all Function Diagrams that assemble the standard Function Blocks into the safety-related logic for a TELEPERM XS project. Main protective measures are:

- Restricted access to the TELEPERM XS Application Code development tools and code files,
- Automated code generation (using only a qualified and verified version of the SPACE Engineering Tool and Function Block Library),
- Independent validation testing, and
- Software file identification by checksums.

Each software development step is followed by an internal design review. Development document reviews are performed by the independent V&V group, which also performs the validation testing. Sections 5.0 and 6.0 of this Annex are applicable to the TELEPERM XS Application Software development work performed by AREVA NP Inc. in Alpharetta, Georgia.

Section 7.0 of this Annex describes a number of TELEPERM XS System features that minimize or eliminate software and data security vulnerabilities. Section 8.0 of this Annex provides an overall conclusion based on the assessment of various scenarios for unauthorized activities by a user or developer for the TELEPERM XS Software development processes.

3.0 TELEPERM XS System Platform Software Development Lifecycle

The phase model for the development of TELEPERM XS System Software is described by TELEPERM XS Phase Model engineering procedures, as noted in Section 3.2.1.2 of the TELEPERM XS Topical Report. The TELEPERM XS System Software is developed in accordance with the TELEPERM XS Software V&V procedure. This section summarizes the main phases to the degree relevant for the topic of unauthorized activities by a user or developer.

The first development phase comprises the elaboration of a requirement specification or frame requirement specification according to the scope of the software component. During this phase all functional requirements and boundary conditions to be met by the software component are stated and identified. Each requirements specification document is structured as stipulated by the TELEPERM XS Software Requirements Specifications engineering procedure. Each requirement must be precise and traceable to subsequent development phases.

The requirement specification phase is completed by a review according to the TELEPERM XS Reviews engineering procedure. During a review, the peer reviewers

check whether the requirements are unambiguous, non contradictory, etc. and whether any requirements are missing.

Based on the requirement specification, a design description or preliminary design description is elaborated according to the TELEPERM XS Software Design Descriptions engineering procedure. A design description introduces design entities, design attributes and the main implementation entities (e.g., the main data structures and functions).

The design description phase is completed by a review that covers the internal consistency of the design description document and the transition from the requirements phase.

The main part of the implementation description is extracted from the source code and source code comments. The implementation description of a TELEPERM XS software component that is subject to external qualification additionally contains a manually rendered part with the content structure defined by TELEPERM XS Software Implementation engineering procedure. These documents also contain a section entitled "Identification of derived software components" that lists the details of the shipped product. For each shipped file the following details are specified: name, checksum, size, and optionally date/version.

The source code adheres to a coding style guide as defined by the TELEPERM XS Software Programming Guidelines engineering procedure.

The source code and objects derived from the source code are administered with a Software Configuration Management (SCM) system. Access to the SCM system is restricted to software developers. The creation and modification of any files or directories is tracked.

The developer documentation is created, reviewed, and released in the document management system of AREVA NP GmbH. The document management system is

maintained by administrators of the AREVA NP GmbH Information Technology department and not by the SCM system administrators. During the creation of a document the document owner assigns roles relevant during the document lifecycle (i.e., editor, reviewers, internal releaser, and external releaser). Only read-only access is provided to released documents, even to the document owner. The document owner may create a new document index (subsequent version of the document).

The third party reviews associated with software requirements and software design descriptions are described in Sections 4.1 and 4.2 of this Annex. AREVA NP GmbH does not define any particular review requirements or methodologies. Instead, the third party organizations conduct independent reviews to ensure that the software was properly developed and conforms to requirements. As a general rule, the third part organization verifies that the software satisfies the specified requirements and contains no other code (i.e., a one to one verification) and also validates that the software meets the specified functional requirements. The third party organization may use various techniques and tools to identify changes to software in later versions and develop the necessary regression tests. The purpose is always to verify that any modifications to the software have not caused unintended effects and that the system or component still complies with its specified requirements.

The TELEPERM XS System Software has substantial operating experience (i.e., more than a decade of experience with the platform installed in 39 units at 24 plant sites located in 11 countries and utilizing 11 different reactor designs) without any reported software and data security problems or indications of software tampering.

4.0 Analysis of the Potential for Unauthorized Activities by User or Developer during TELEPERM XS System Platform Software Development

The possible user or developer activities with regard to the above mentioned development phases are listed below:

Security-Related Information Withheld in Accordance with 10 CFR 2.390

The analysis which follows, postulates that potential intruders have a profound knowledge of the TELEPERM XS products (hardware, software, network protocols) and the I&C projects (overall process engineering functionality, project-specific network topology, redundancy concept etc.). No type of unauthorized activities by a user or developer is excluded from the analysis. Especially, the intentional introduction of unsafe designs and vulnerable programming constructs is not excluded.

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld
in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld
in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

5.0 TELEPERM XS Application Software Development Lifecycle

The project phases used for TELEPERM XS projects correlate to the project phases described in Regulatory Guide 1.152 and the V&V process used for TELEPERM XS projects, as shown in Table 4-2.

The requirements phase comprises the elaboration of a TELEPERM XS Application Software Requirements Specification (SRS) according to the project system and functional requirements. During this phase, all functional requirements and boundary conditions to be met by the project-specific TELEPERM XS Application Software are stated and identified. Each requirements specification document is structured in accordance with the requirements of IEEE Std 730-2002 and IEEE Std 830-1993. This

document is reviewed for adequacy in meeting those standards by an independent reviewer and then for traceability of requirements from the customer's specification and industry standards. The independent V&V group performs the Software Requirements Phase Traceability Analysis to show forward and backward traceability between requirements, design, implementation, and test documents. They also perform a Software Requirements Evaluation to verify that the Software Requirements Specification adequately defines the software requirements necessary to perform intended functions.

The design phase comprises the elaboration of a TELEPERM XS Application Software Design Description (SDD) based on the SRS. The SDD is comprised of various views of the TELEPERM XS Application Software, including an overview of the system architecture, and a top-level presentation of the important system functions. The SDD is reviewed for adequacy in meeting those recommendations by an independent reviewer and then for traceability of requirements from the SRS. The independent V&V group performs the Software Design Phase Traceability Analysis to show forward and backward traceability between the SRS and SDD. They also perform a Software Design Evaluation to verify that the software requirements are adequately addressed and built into the design.

The implementation phase of the TELEPERM XS Application Software Code includes engineering, generation, and download steps. The design documented in the SDD is implemented in the Engineering Database using the SPACE Engineering Tool and associated Function Block Library. The Application Code is checked by a visual check of SPACE diagrams by an independent reviewer and engineering checks of the functionality in the simulation test environment. The next step consists of generation of the C source code and compilation by using the SPACE Engineering Tool Code Generators in accordance with the project Software Generation and Download Procedure. The independent V&V group performs the Software Implementation Phase Traceability Analysis to show forward and backward between the Application Software Code (SPACE Function Diagrams) and the SDD. They also perform the Source Code

and Source Code Documentation Evaluation to ensure that the software design is correctly translated into the Application Software Code documents by verifying that:

- SPACE Function Diagrams were prepared using the recommendations, conventions, and design constraints given in the TELEPERM XS Function Block Manual corresponding to the version of the TELEPERM XS System Software used on the project,
- Correct versions and revisions of SPACE Function Diagrams were used to assemble the Application Software release for each TELEPERM XS subsystem,
- Procedure used for generation of the Application Software release was followed and is consistent with the Software Configuration Management Plan,
- Qualified versions of SPACE Engineering Tool Code Generators have been used, and
- Only qualified TELEPERM XS System Software components (including the Function Block Library) have been used.

The TELEPERM XS Application Software code and objects derived from the source code are administered in accordance with the Software Library and Control procedure. Access to the Software Library is restricted to software developers. The creation and modification of any files or directories are tracked.

The TELEPERM XS Application Software documentation is created, reviewed, and released in the Records Management System of AREVA NP Inc. The documentation includes the Code Configuration Documents identifying the TELEPERM XS Application Software Releases. The Records Management System is maintained by administrators of the AREVA NP Inc. Information Management Systems department and not by the software engineers or the TELEPERM XS Project Software Librarian. During the creation of a document, the document owner assigns roles relevant during the document lifecycle (i.e., editor, reviewers, internal releaser, and external releaser). Only read-only access is provided to released documents, even to the document owner. The

document owner may create a new document index (subsequent version of the document).

The test phase of the TELEPERM XS Application Software Code development includes software integration and system validation testing performed by the independent V&V group. The independent V&V group performs the Software Test Phase Traceability Analysis to show forward and backward traceability between requirements, design, implementation, and test documents. A Physical Configuration Audit is performed after the TELEPERM XS Software to be used during Factory Acceptance Test is installed on the target hardware and before the start of the independent validation testing. The Physical Configuration Audit is executed under the lead of an independent certified lead auditor and may consist of members from the independent V&V group and other QA personnel. The Physical Configuration Audit verifies that the correct versions of the TELEPERM XS Application Software were used in the validation testing, as specified in the Software Configuration Management Plan.

The AREVA NP Inc. QA group performs a functional audit once the independent V&V group completes its test phase summary activity report and prior to the system being released to the customer for site acceptance testing. This functional audit assures that all of the processes detailed in the Software Verification and Validation Plan and Software QA Plan were followed. The independent QA functional audit and oversight confirms that the final product meets the regulatory and customer requirements adequately and no unauthorized requirements were introduced.

The TELEPERM XS project Installation and Checkout phase consists of verifying the configuration of software installed on the system released to the customer for site acceptance testing. The independent V&V group verifies that all required Application Software packages are installed and that the TELEPERM XS Application Software versions are current and controlled in accordance with the Software Configuration Management Plan.

6.0 Analysis of the Potential for Unauthorized Changes by User or Developer during TELEPERM XS Application Software Development

The possible unauthorized activities by a user or developer with regard to the above mentioned development phases are listed below:

Security-Related Information Withheld in Accordance with 10 CFR 2.390

The analysis which follows, postulates that potential intruders have a profound knowledge of the TELEPERM XS Application Software (including development tools) and the I&C projects (overall development process and project-specific information). No types of unauthorized modification, or intentional introduction of unsafe designs, or vulnerable programming constructs by a user or developer are excluded from the analysis.

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld
in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

Security-Related Information Withheld in Accordance with 10 CFR 2.390

7.0 TELEPERM XS System Design Features that Minimize Software and Data Security Vulnerabilities

The TELEPERM XS System has a number of features that minimize or eliminate software and data security vulnerabilities.

The TELEPERM XS System is designed for strictly cyclic signal processing. No process interrupts are used. This design feature increases the predictability of the software and ensures that operation is unaffected by plant events, as signaled through the data processed. Watchdog timers are used to monitor processor operation. As such, the TELEPERM XS software strategy eliminates vulnerabilities to certain software vulnerabilities (e.g., interrupt-triggered undocumented software).

The TELEPERM XS System is designed as an asynchronous operating system. The TELEPERM XS System does not utilize a real-time clock to synchronize redundant processors. This design feature eliminates synchronization failure modes and enhances system reliability. As such, the TELEPERM XS software strategy eliminates vulnerabilities to certain software vulnerabilities (e.g., time bomb undocumented software).

The TELEPERM XS System uses only static memory allocation. Each variable in the application program has a permanent dedicated place in memory defined at the time of code generator. This feature eliminates failure modes associated with memory conflicts caused by dynamic memory allocation. As such, the TELEPERM XS software strategy eliminates vulnerabilities to certain software vulnerabilities (e.g., viruses and worms).

The TELEPERM XS Software is custom and proprietary software. The use of the TELEPERM XS System Software is limited to nuclear applications. Consequently, the software does not have the same software and data security vulnerabilities as open-access industrial software or widely distributed public software.

The TELEPERM XS Safety System does not use TCP/IP software or message routing services, which are software features that are commonly exploited in unauthorized software and data events. As such, the TELEPERM XS software strategy eliminates important transmission vectors for unauthorized external software and data events (e.g., control system intrusions, malevolent code infection, and denial of service events).

The TELEPERM XS System is designed to execute program code from FEPROM. The program code on the FEPROM can only be changed from the TELEPERM XS Service Unit. The TELEPERM XS System is designed without the use of external data storage media. Consequently, the TELEPERM XS System has no memory storage media where malevolent code can hide or replicate. These design features eliminate the vulnerability to unauthorized software and data events caused by malevolent code. The integrity of the code installed on the FEPROMs is routinely checked for integrity as part

of the cyclic test features. This design feature ensures that any damaged code is promptly identified and isolated.

Application Software is generated for each TELEPERM XS computer using the TELEPERM XS automated code generator tool. The code generation tool assigns a unique ID for each TELEPERM XS computer, configuration data for all input/output drivers, and the complete lists of all messages to be sent or received by the TELEPERM XS computers. Communication between TELEPERM XS computers is designed with logical point-to-point connections. Separate logical MicroNET communication channels are provided for each message. The TELEPERM XS communication protocol does not use acknowledges by the receiver. The TELEPERM XS System is designed for strictly cyclic data transfer via network connections, using a predefined package size and a constant bus load, with checksum and message age monitoring. The TELEPERM XS System is designed to operate with a fixed cycle time. Each TELEPERM XS computer reads input data and writes output data once per cycle. This design feature eliminates the vulnerability to network data storms or unauthorized software and data events (i.e., denial of service events). The TELEPERM XS message transfer protocol, when coupled with other design features, eliminates additional software and data security concerns (e.g., control system intrusions and spoofing).

The MSI is the only physical network connection that exists between the TELEPERM XS safety function processors and non-safety systems (e.g., TELEPERM XS Gateway or TELEPERM XS Service Unit). The TELEPERM XS Gateway is also isolated form the TELEPERM XS Service unit by a MSI or the use of a one-way hardware device. No TCP/IP or message routing services are installed on the MSI. These design features eliminate important transmission vectors for unauthorized external software and data events.

8.0 Conclusion

The software development and deployment phases for TELEPERM XS System and Application Software exhibit different degrees of susceptibility to unauthorized changes by a user or developer. The analysis of the previous sections addressed all phases to a detail commensurate with the potential for unauthorized changes by a user or developer. The measures imposed by the TELEPERM XS engineering procedures assure that all countermeasures against unauthorized changes at any time in the TELEPERM XS Software development life cycle are adequately addressed.

Appendix D – TELEPERM XS Platform and Procedure Change Evaluation Process

NRC approved the initial set of TELEPERM XS development procedures and hardware and software modules as part of their approval of the TELEPERM XS Topical Report. NRC may have also approved additional hardware and software modules as part of their approval of license amendment requests for TELEPERM XS projects or through Inspection, Test, and Analysis Acceptance Criteria/Design Acceptance Criteria closure related to new build projects.

D.1 Criteria for Use of Future TELEPERM XS Platform Changes

AREVA NP can use newer TELEPERM XS hardware and software modules provided that the following criteria are met:

- Changes to the TELEPERM XS platform do not modify or eliminate the key design principles, as described in the TELEPERM XS Topical Report. (See question 1 in Section D.2)
- Changes to the TELEPERM XS platform do not modify or eliminate the key processing features, as described in the TELEPERM XS Topical Report. (See question 2 in Section D.2)
- Changes to the TELEPERM XS platform do not modify or eliminate the key communication independence features, as described in the TELEPERM XS Topical Report. (See question 3 in Section D.2)
- Changes to the TELEPERM XS platform (hardware and software modules) do not result in more than a minimal increase in the likelihood of occurrence of a malfunction of the TELEPERM XS software. (See questions 4 and 5 in Section D.2)
- Changes to the TELEPERM XS hardware do not result in more than a minimal increase in the consequences of a malfunction in the TELEPERM XS System. (See question 6 in Section D.2)

- Changes to the TELEPERM XS hardware do not create a possibility for a malfunction of a TELEPERM XS System with a different result. (See question 7 in Section D.2)

AREVA NP can use TELEPERM XS hardware and software modules developed with modified TELEPERM XS development procedures provided that the following criterion is met:

- Changes to the TELEPERM XS development procedures do not result in a reduction in the quality methods described in the TELEPERM XS Topical Report. (See question 8 in Section D.2)

TELEPERM XS platform and procedure changes that satisfy these criteria conform to the NRC approval basis for the TELEPERM XS Topical Report.

AREVA NP can develop project-specific TELEPERM XS Application Software with modified TELEPERM XS project development procedures provided that the following criterion is met:

- Changes to the TELEPERM XS project development procedures do not result in a reduction in the quality methods described in the TELEPERM XS Software Program Manual Topical Report. (See question 9 in Section D.2)

TELEPERM XS project development procedure changes that satisfy this criterion conform to the NRC approval basis for the TELEPERM XS Software Program Manual Topical Report.

The TELEPERM XS system, hardware, and software change analysis criteria have been developed to reflect their use for generic platform based on key TELEPERM XS system features described in the TELEPERM XS Topical Report.

D.2 Evaluation Process for Use of Future TELEPERM XS Platform Changes

AREVA NP will evaluate the TELEPERM XS hardware and software modules, as well as the associated development procedures, used for a TELEPERM XS project that have not been previously reviewed and approved by NRC (i.e., in the TELEPERM XS Topical Report, the TELEPERM XS Software Program Manual Topical Report, or project-specific licensing approvals), using the following review guidelines.

1. Does the change to the TELEPERM XS platform (i.e., hardware or software modules) modify or eliminate the key design principles, as described in the TELEPERM XS Topical Report?

   These areas should be addressed in responding to the question:

   (a) Does the change modify or eliminate the use of random access memory for the storage of program and system variables, flash erasable programmable read only memory for storing program code, or electrical erasable programmable read only memory for storing application program data?

   (b) Does the change modify or eliminate the use of a modular approach for system software that separates the runtime environment, communication software, hardware-specific software, and application software?

   (c) Does the change modify or eliminate the way the application software is developed to perform the plant-specific safety-related functions using qualified function block modules which are grouped into function diagram modules (e.g., eliminate the use of the SPACE engineering tool for automatic code generation)?

   (d) Does the change modify or eliminate the requirement that application software be generated by TELEPERM XS SPACE engineering tool,

which uses qualified software modules from the function block library to construct application software?

(e) Does the change modify or eliminate the cyclic self-testing performed to monitor system operation?

(f) Does the change modify or eliminate the use of watchdog timers to monitor processor performance?

(g) Does the change modify or eliminate the access control features that are used to protect the safety processors from unauthorized interventions from the Service Unit?

2. Does the change to the TELEPERM XS platform (i.e., hardware or software modules) modify or eliminate the key processing features, as described in the TELEPERM XS Topical Report?

These areas should be addressed in responding to the question:

(a) Does the change modify or eliminate the use of a deterministic operating system with a strictly cyclic processing of application software?

(b) Does the change modify or eliminate the way the operating system operates in an asynchronous manner?

(c) Does the change modify or eliminate the use of only static memory allocation?

(d) Does the change add the use of any process-driven interrupts?

(e) Does the change add the use of long-term data storage within the TELEPERM XS System?

(f) Does the change add any use of external data storage media within the TELEPERM XS System?

(g) Does the change introduce a real-time clock into the TELEPERM XS System?

3. Does the change to the TELEPERM XS platform (i.e., hardware or software modules) modify or eliminate the key communication independence features, as described in the TELEPERM XS Topical Report?

These areas should be addressed in responding to the question:

(a) Does the change modify or eliminate the method of cyclic data transfer via network connections, use of predefined package size for messages, or constant bus load operation?

(b) Does the change modify or eliminate the use of individual dual port random access memory buffers for each message with separation of the data flow for sending and receiving?

(c) Does the change modify or eliminate the checks performed on all received messages for message authentication, validity checks, and message age monitoring?

4. Does the change to the TELEPERM XS software result in more than a minimal increase in the likelihood of occurrence of a malfunction of the software module?

These areas should be addressed in responding to the question:

(a) Was the software module developed under a different process than the 10 CFR Part 50 Appendix B quality assurance program using the life-cycle development process described in the TELEPERM Topical Report?

(b) Was the software module developed to different programming standards than the TELEPERM programming guidelines?

(c) Does the change to the software module introduce different behavior or potential failure modes that could affect the safety-related TELEPERM design functions?

(d) Was the safety-related software module qualified under a different process than by an independent third party reviewer, as required by the life-cycle development process described in the TELEPERM Topical Report?

5. Does the change to the TELEPERM XS hardware result in more than a minimal increase in the likelihood of occurrence of a malfunction of the component?

These areas should be addressed in responding to the question:

(a) Was the hardware module developed under a different process than the 10 CFR Part 50 Appendix B quality assurance program and development process described in the TELEPERM Topical Report?

(b) Is the component qualified to lesser qualification requirements for seismic, environmental, electromagnetic compatibility, and electrical isolation than those described in the TELEPERM XS Topical Report or subsequently endorsed by NRC Regulatory Guides?

(c) Did the design analysis for the change fail to consider the electrical loads associated with the component?

(d) Did the design analysis for the change fail to consider the thermal loads associated with the component?

(e) Did the design analysis for the change fail to consider applicable requirements for separation, independence, and grounding?

(f) Did the design analysis for the change fail to consider applicable industry and regulatory standards?

6. Does the change to the TELEPERM XS hardware result in more than a minimal increase in the consequences of a malfunction?

   These areas should be addressed in responding to the question:

   (a) Does the change introduce failure modes that were not present or analyzed in the previous version that result in a more severe condition and/or create failures that are not detectable during operation or through periodic testing?

7. Does the change to the TELEPERM XS hardware create a possibility for a malfunction of a TELEPERM XS System with a different result?

   These areas should be addressed in responding to the question:

   (a) Is the component qualified to lesser qualification requirements for seismic, environmental, electromagnetic compatibility, and electrical isolation than those described in the TELEPERM XS Topical Report?

   (b) Does the component combine previously separate functions into one digital device such that a failure creates a result not bounded by the results of malfunctions previously considered in the TELEPERM XS Topical Report?

   (c) Does the modified hardware component have the same failure mode on loss of power as the component being replaced? If the failure mode is different, are the consequences increased beyond what was evaluated previously for the TELEPERM XS System?

   (d) Is the response of the component on restoration of power different from that of the component being replaced? If so, are the consequences bounded by other TELEPERM XS System features?

   (e) Does the component reset to operating parameters and settings established for the specific system, or does it go to a different default set of parameters when the component is reset?

(f) Does the component introduce any new or different system-level failure modes and effects that are not bounded by the results previously considered in the TELEPERM XS Topical Report?

8. Does the change to TELEPERM XS development process procedure result in a reduction in the quality methods described in the TELEPERM XS Topical Report?

These areas should be addressed in responding to the question:

(a) Does the procedure change involve a reduction to any element of the development process procedures that are described in the TELEPERM XS Topical Report which are used to ensure platform hardware or software quality?

(b) Does the change involve a reduction in the level of commitment to any industry standards that form the basis for NRC approval in the TELEPERM XS Topical Report?

9. Does the change to TELEPERM XS project development procedure result in a reduction in the quality methods described in the TELEPERM XS Software Program Manual Topical Report?

These areas should be addressed in responding to the question:

(a) Does the procedure change involve a reduction to any element of the development process procedures that are described in the TELEPERM XS Software Program Manual Topical Report which are used to ensure project-specific Application Software quality?

(b) Does the change involve a reduction in the level of commitment to any industry standards that form the basis for NRC approval in the TELEPERM XS Software Program Manual Topical Report?

If questions 1 through 8 are answered "no" that means that the criteria listed in Section D.1 have been met and the new or modified TELEPERM XS hardware or software

module can be used for a TELEPERM XS project. The records of these evaluations are maintained in accordance with Section D.3 and available for audit by NRC.

If any of the questions 1 through 8 are answered "yes" that means that one or more of the criteria listed in Section D.1 have not been met and the new or modified TELEPERM XS hardware or software module cannot be used for a TELEPERM XS project without specific NRC review and approval.

Similarly, if question 9 is answered "no" that means that the criterion listed in Section D.1 has been met and the modified TELEPERM XS project development procedure can be used for a TELEPERM XS project. The records of these evaluations are maintained in accordance with Section D.3 and available for audit by NRC.

If question 9 is answered "yes" that means that the criterion listed in Section D.1 have not been met and the modified TELEPERM XS project development procedure cannot be used for a TELEPERM XS project without specific NRC review and approval.

The TELEPERM XS system, hardware, and software change evaluation process is based on insights from the available industry guidance: NEI 96-07 (Reference 41), as endorsed by Regulatory Guide 1.187 (Reference 15) and EPRI TR-102348 (Reference 22), as endorsed by NRC Regulatory Issue Summary 2002-22 (Reference 45). The change evaluation process for the TELEPERM XS platform has been customized to reflect its use for generic platform based on key TELEPERM XS system features described in the TELEPERM XS Topical Report.

D.3 Change Evaluation Records

AREVA NP will maintain records of changes to the TELEPERM XS platform or project development procedures used for TELEPERM XS projects in the U.S. in accordance with the AREVA NP Records Management Program. These records must include a written evaluation which provides the bases for the determination that the TELEPERM XS platform or project development procedure change met the criteria listed in Section

D.1. The basis for the answer to each question will be documented in sufficient detail to support a review by independent parties (e.g., customer or QA) or audits by NRC. In cases where information or discussion applies to more than one question, the applicable information or discussion shall be documented under each question to provide a complete basis for the answer to each question.