

<i>UF/NRE</i> <i>UFTR</i>	<i>QUALITY ASSURANCE DOCUMENT</i>	<i>Project ID: QA-1</i>	
		<i>Revision 1</i>	<i>Copy 1</i>
		<i>Page 1 of 40</i>	

Project Title: *UFTR DIGITAL CONTROL SYSTEM UPGRADE*

UFTR-QA1-01, Software Quality Assurance Plan (SQAP)

Prepared by,

Dr. Gabriel Ghita

[Handwritten Signature] (Signature)

Date: *4/12/10*

Reviewed by,

Prof. Mark Harrison

For M. Harrison *[Handwritten Signature]* (Signature)

Date: *4/12/10*

Approved by,

Prof. Alireza Haghghat

[Handwritten Signature] (Signature)

Date: *4/12/10*

UF/NRE UFTR	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 4 of 40</i>

Table of Contents

1. Purpose	7
2. Reference Documents	8
2.1 UFTR Documents	8
2.2 AREVA NP Inc. Documents	8
2.3 Industry Standards	8
2.4 NRC Documents	9
3. Management	10
3.1 Organization	10
3.2 Tasks	10
3.2.1 Basic Design Phase Tasks	10
3.2.2 Detailed Design Phase Tasks	11
3.2.3 Testing Phase Tasks	12
3.3 Roles and responsibilities	13
3.4 Quality Assurance Estimated Resources	13
3.5 Verification and Validation (V&V)	13
3.5.1 Activities	13
3.5.2 Technical Independence	14
3.5.3 Managerial Independence	14
3.5.4 Financial Independence	14
3.5.5 Escalation Protocol	14
4. Documentation	15
4.1 Purpose	15
4.2 Minimum Documentation Requirements	15
4.2.1 Software Requirement Specification (SRS)	15
4.2.2 Software Design Description (SDD)	15
4.2.3 Application Software Code Document	16
4.2.4 Verification and Validation Plan	16
4.2.5 Verification Results Report and Validation Results Report	16
4.2.6 User Documentation	17
4.2.7 Software Configuration Management Plan (SCMP)	17
4.3 Other Documentation	17
5. Standards, Practices, Conventions, and Metrics	19

UF/NRE UFTR	Prepared by		Reviewed by		QA-1, UFTR-QA1-01	
	Name:		Name:		Revision 1	Copy 1
	Date :	Initials:	Date :	Initials:	Vol. 1	Page 5 of 40

- 5.1 Documentation Standards.....19**
- 5.2 Design Standards.....19**
- 5.3 Coding Standards.....19**
- 5.4 Commentary Standards20**
- 5.5 Testing Standards and Practices.....20**
- 5.6 Software Quality Assurance Product and Process Metrics20**
- 6. Reviews and Audits.....22**
 - 6.1 Reviews.....22**
 - 6.1.1 Software Specification Review (SSR)22**
 - 6.1.2 Preliminary Design Review (credited for the Architecture Design Review)22**
 - 6.1.3 Detailed Design Review22**
 - 6.1.4 Verification and Validation Plan Review (SVVPR)23**
 - 6.1.5 Managerial Reviews23**
 - 6.1.6 Software Configuration Management Plan Review23**
 - 6.1.7 Post-Implementation Review23**
 - 6.2 Audits23**
 - 6.2.1 In-Process Audits.....23**
 - 6.2.2 Physical Audits24**
 - 6.2.3 Functional Audits24**
 - 6.2.4 Other Reviews and Audits - Software Process Audits24**
- 7. Testing.....25**
- 8. Problem Reporting and Corrective Action26**
- 9. Tools, Techniques, and Methodologies.....27**
 - 9.1 Methodology for Generating the SRS27**
 - 9.2 Tools for Generating the SDD27**
 - 9.3 Tools for the Specification and the Generation of the Application Software27**
 - 9.4 Tools for Software Simulation Testing27**
 - 9.5 Tools for Verification and Validation28**
- 10. Media Control29**
 - 10.1 Media Control for the SDD.....29**
 - 10.2 Media Control for Application Software.....29**
 - 10.3 Code Control of TXS System Software.....29**

UF/NRE UFTR	Prepared by		Reviewed by		QA-1, UFTR-QA1-01	
	Name:		Name:		Revision 1	Copy 1
	Date :	Initials:	Date :	Initials:	Vol. 1	Page 6 of 40

11. Supplier Control.....	30
12. Records Collection, Maintenance, and Retention	31
13. Training	32
14. Risk Management.....	33
15. Glossary	34
15.1 Definitions.....	34
15.2 Abbreviations and Acronyms	39
16. SQAP Change Procedure and History	40

UF/NRE UFTR	Prepared by		Reviewed by		QA-1, UFTR-QA1-01	
	Name:		Name:		Revision 1	Copy 1
	Date :	Initials:	Date :	Initials:	Vol. 1	Page 7 of 40

1. Purpose

The UFTR is planning to use the TELEPERM XS (TXS) platform and application software to provide appropriate Reactor Protection System (RPS) functions.

This Software Quality Assurance Plan (SQAP) provides the necessary measures to make sure that the developed TXS Application Software conforms to established technical requirements, rules, and standards. The Plan fulfills the requirements for a SQAP and conforms to IEEE Std. 730-2002, "IEEE Standard for Software Quality Assurance Plans," ~~20/19/~~, describing the tools to be used and methodologies to be followed in developing and maintaining of TXS Application Software.

Together with UFTR "Software Verification and Validation Plan (SVVP)," ~~5/~~, the purpose of this plan is to:

- Detect and eliminate design errors in design phases
- Enhance the quality and reliability of the I&C System

This plan covers the design, testing and documentation phases for the TXS Application Software, TXS Graphic Service Monitor (GSM) Application Software, and TXS Qualified Display System (QDS) Application Software, as described in the UFTR "Quality Assurance Project Plan (QAPP)," ~~3/~~.

Software elements produced in the process of Quality Assurance are as follows:

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

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 8 of 40</i>

2. Reference Documents

2.1 UFTR Documents

- /1/ UFTR-QAP, "Quality Assurance Program (QAP)"
- /2/ UFTR-QAP-01-P, "Conduct of Quality Assurance"
- /3/ UFTR QA1-QAPP, "Quality Assurance Project Plan (QAPP)"
- /4/ UFTR-QA1-02, "Software Configuration Management Plan (SCMP)"
- /5/ UFTR-QA1-03, "Software Verification and Validation Plan (SVVP)"
- /6/ UFTR-QA1-05, "Software Safety Plan (SSP)"
- /7/ UFTR-QA1-06.1, "Software Test Plan – SIVAT Plan"
- /8/ UFTR-QA1-06.2, "Factory Acceptance Plan – FAT Plan"
- /9/ UFTR-QA1-09, "Software Operations and Maintenance Plan"
- /10/ UFTR-QA1-10, "Software Training Plan"
- /11/ UFTR-QA1-12, "Software Reviews and Audits"
- /12/ UFTR-QA1-100, "Functional Requirements Specifications (FRS)"
- /13/ UFTR-QA1-102.1, "ID Coding Concept"
- /14/ UFTR-QA1-105, "Cyber-Security"
- ~~UFTR-QA1-108, "Requirements Traceability Matrix (RTM)"~~
- /15/ UFTR-QA1-109, "Software Library and Control"

2.2 AREVA NP Inc. Documents

- /16/ AREVA NP Inc. Document No., 38-1288541-00, Topical Report EMF-2110(NP) (A) Revision 1, "TELEPERM XS: A Digital Reactor Protection System"
- /17/ AREVA NP Inc. Document No., 38-9033245-000, "Safety Evaluation by the Office of Nuclear Reactor Regulation Siemens Power Corporation Topical Report EMF-2110(NP), "TELEPERM XS: A Digital Reactor Protection System, Project No. 702," March 5, 2000"

2.3 Industry Standards

- /18/ IEEE Std 610.12-1990, "IEEE Standard Glossary of Software Engineering Terminology"
- /19/ IEEE Std 730-2002, "IEEE Standard for Software Quality Assurance Plans"
- /20/ IEEE Std 828-1990, "IEEE Standard for Software Configuration Management Plans"
- /21/ IEEE Std 830-1998, "IEEE Recommended Practice for Software Requirements Specifications"
- /22/ IEEE Std 1016-1998, "IEEE Recommended Practice for Software Design Descriptions"
- /23/ IEEE Std 1028-1997, "IEEE Standard for Software Reviews"
- /24/ IEEE Std 1063-2001, "IEEE Standard for Software User Documentation"

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 9 of 40</i>

/25/ IEEE Std. 1012-1998, "IEEE Standard for Software Verification and Validation"

2.4 NRC Documents

/26/ Regulatory Guide 1.168, Rev. 1, February 2004, "Verification, Validation, Reviews, and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants"

/27/ Regulatory Guide 1.169, September 1997, "Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants"

/28/ Regulatory Guide 1.172, Rev. 0, September 1997, "Software Requirements Specifications for Digital Computer Software Used In Safety Systems of Nuclear Power Plants"

<i>UF/NRE UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 10 of 40</i>

3. Management

3.1 Organization

A description of each major element of the project organizational structure and its responsibilities is found in the UFTR QAPP, /3/, along with a discussion of the major roles and responsibilities of each element.

Preparing and maintaining the SQAP is the responsibility of the Project Coordinator. The UFTR Reactor Manager shall approve the SQAP.

Organizationally, verification of the implementation of quality assurance requirements shall be performed by the Project Auditor in accordance with the UFTR "Quality Assurance Program (QAP)", /1/. The Project Coordinator shall ensure that the software and associated documentation have been developed in accordance with the SQAP. The Project Manager ensures project work activities are accomplished in accordance with this Plan.

The Independent Verification and Validation (IV&V) Lead shall be responsible for V&V activities including preparation and review of V&V products and documentation in accordance with the UFTR SVVP, /5/.

3.2 Tasks

The sequence of the tasks organized by phase and covered by this plan is listed below. The tasks describe the complete program intended for development and V&V of the Application Software. An explanation of the initiating criteria and outputs for each task is given in the following sections.

3.2.1 Basic Design Phase Tasks

A Software Requirements Specification (SRS) shall be prepared for each applicable software product. The SRS shall be developed based on the Functional Requirements Specification (FRS), /12/, the hardware interface description, and other applicable documents. The preparation of the SRS shall be accomplished by the Software Development Group. The document shall be independently reviewed by another member of the Software Development Group and approved by the Software Development Lead. The SRS is described further in Section 4.2.1.

The IV&V Group shall review the SRS. The SRS shall be reviewed each time it is updated. The output from the review shall be verification review comments, which shall be documented in the Activity Summary Report for the current V&V phase and transmitted to the development group. The report shall identify all the deficiencies discovered during the review. The IV&V SRS review report shall be independently reviewed by another member of the IV&V Group. The report shall be approved by the IV&V Lead. The IV&V SRS review shall be performed in accordance with Section 3.5.

A SVVP shall be prepared to describe the methods to be used to verify and validate the software. The SVVP shall be updated and tailored to each of the

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 11 of 40</i>

software development phase products as they are produced. The SVVP shall be prepared by a member of the IV&V Group and independently reviewed by another member of the group. The plan shall be approved by the IV&V Lead and by the Quality Assurance Project Auditor. The SVVP is described further in Section 4.2.4.

A Software Verification and Validation Plan Review (SVVPR) shall be performed after the initial SVVP is produced. Subsequent reviews shall occur if needed as the IV&V work progresses. The output of this task will be updates to the SVVP. The SVVPR shall be prepared by the reviewers and shall be approved by the IV&V Lead. The input to the SVVPR shall be the SVVP. The SVVPR is further described in Section 4.2.4.

3.2.2 Detailed Design Phase Tasks

A Software Design Description (SDD) shall be prepared as a product of the design process. The TXS Application Software SDD consists of the complete set of function diagrams developed using the FunBase tool. The input for the TXS Application Software SDD is the TXS Application Software SRS. Since GSM and QDS Application Software are primarily composed of individual graphical dialogue screens and scripts, an SDD is not required. All necessary information for documenting these software products shall be included in the corresponding GSM code documents. The TXS Application Software SDD is discussed further in Section 4.2.2.

The IV&V Group shall review the TXS Application Software SDD. The SDD shall be reviewed each time it is updated. The output from the review shall be verification review comments, which shall be documented in the Activity Summary Report for the current V&V phase and transmitted to the development group. The report shall identify all the deficiencies discovered during the review. The review process may require several iterations. The IV&V SDD review report shall be independently reviewed by another member of the IV&V Group. The report shall be approved by the IV&V Lead. The IV&V SDD review shall be performed in accordance with Section 3.5.

A software safety analysis is performed in accordance with UFTR "Software Safety Plan (SSP)" /6/.

For the TXS Application Software, the code is generated by the TXS object-oriented automated code generation tool (SPACE). As such, the Software Development Group does not create code. The use of the automatic code generator eliminates a potential source of errors by eliminating the human interface between I&C function development and code generation. The TXS Application Software SDD logic diagrams are manually entered into the SPACE function diagrams. The automatic generation of code from the SPACE function diagrams eliminates the need for the manual creation of code which removes the potential for human error.

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 12 of 40</i>

An Application Software Code Document is generated by the Software Development Group and is a graphical representation of the TXS RPS Application Software utilizing the TXS engineering tool SPACE.

For the TXS GSM and QDS Application Software, the code is generated using a combination of manually created scripts and Dialogue Screens, which are created by the Qt Designer tool utilizing special GSM and QDS libraries.

The IV&V Group shall perform a Software Implementation Review. This shall be the V&V review of the TXS Application Software Code Document. The Code Document shall be reviewed each time it is updated. The output from the review shall be verification review comments, which shall be documented in the Activity Summary Report for the current V&V phase and transmitted to the development group. The report shall identify all the deficiencies discovered during the review. The review process may require several iterations. The IV&V Software Implementation review report shall be independently reviewed by another member of the IV&V Group. The report shall be approved by the IV&V Lead. The IV&V implementation review shall be performed in accordance with Section 3.5.

A Source Code Review (SCR) will not be performed because the software safety coding analysis is done through the use of software tools. The TXS software tool SPACE has been reviewed and accepted by the NRC as stated in the NRC SER, ~~A8/17/~~, and the TXS Topical Report, ~~A7/16/~~.

The correctness of the TXS Application Software in the course of specific projects is ensured by software simulation testing either as an engineering debugging activity or as a formal validation testing with a NRC approved test tool as described in Section 3.2.3.

3.2.3 Testing Phase Tasks

Application Software integration and functional testing can be performed with an NRC approved simulation test tool to satisfy 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. Testing in the simulation environment with a NRC approved simulation tool can serve as module or unit testing (i.e., Function Diagram or Function Diagram Group testing). It can also serve as integration testing of the TXS Application Software (i.e., testing of the Application Software for all TXS modules working together) within the limitations of simulation. Additional testing is performed as part of the Factory Acceptance Testing (FAT) to address the limitations of simulation testing. If IV&V performs SIVAT testing then the Software Development Group does not need to perform SIVAT testing as a part of the design process.

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 13 of 40</i>

If Application Software integration and functional testing is not performed with a NRC approved simulation test tool then the required testing shall be performed in the FAT to satisfy IEEE Std 1012-1998 validation requirements.

The correct implementation of the GSM and/or QDS Application Software shall be validated during FAT by utilizing the target system TXS Application software and hardware.

Simulation Test and FAT plans, test specifications (procedures), and test reports (if generated) shall be prepared by a member of the IV&V Group and independently reviewed by another member of the IV&V Group. These documents shall be approved by the IV&V Lead. The IV&V testing shall be performed in accordance with Section 3.5.

3.3 Roles and responsibilities

Specific assignments for tasks are listed in Section 3.2.

3.4 Quality Assurance Estimated Resources

The activities prescribed by this plan are monitored by a dedicated QA Project Auditor, independent of the System Design & Analysis, Software Development, IV&V, and Hardware & Testing Groups. The QA Project Auditor participates in phase-end software reviews and the documentation review of plans, procedures, and reports to assure adherence to the UFTR QAP, /1/. The QA Project Auditor is responsible for review and approval of the following documents:

- Open Item process,
- Software Library,
- Cyber Security
- Software Safety Plan
- Software Configuration Management Plan
- Software Verification and Validation Plan
- Software Operations and Maintenance Plan

3.5 Verification and Validation (V&V)

V&V tasks shall be performed in accordance with the UFTR SVVP, /5/, by the IV&V Group under the supervision of the IV&V Lead. The Project Manager shall ensure adequate independence between the V&V, and Software Development Group in accordance with IEEE Std. 1012, ~~/26/25/~~, as endorsed by the NRC Regulatory Guide 1.168, ~~/27/26/~~.

3.5.1 Activities

All V&V activities are performed in accordance with the UFTR SVVP, /5/.

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 14 of 40</i>

3.5.2 Technical Independence

The IV&V and Software Development groups shall be independent. Project Management will coordinate issues between the groups.

3.5.3 Managerial Independence

The IV&V shall be directed by the IV&V Lead. This individual will be separate from the Software Development Lead.

3.5.4 Financial Independence

The main portion of the salaries of the members of the IV&V group shall be provided by organizations independent of the UFTR Digital Control Upgrade Project funds

3.5.5 Escalation Protocol

The IV&V issues or concerns shall be identified and documented for Software Development Lead and Staff to resolve in accordance with the SVVP, /5/. IV&V issues or concerns which cannot be resolved at the Software Development and IV&V Lead level are escalated to the Project Coordinator and Project Manager for resolution.

UF/NRE UFTR	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 15 of 40</i>

4. Documentation

4.1 Purpose

The purpose of this section is to:

- Identify the documentation governing the development, V&V, use and maintenance of the Application Software
- List which documents are to be reviewed or audited for adequacy.
- Identify the criteria to which adequacy are to be confirmed.

4.2 Minimum Documentation Requirements

The following minimum documentation produced for Application Software is required:

- Software Quality Assurance Plan (SQAP)
- Software Requirements Specification (SRS)
- Software Design Description (SDD)
- Software Safety Plan (SSP)
- Software Configuration Management Plan (SCMP)
- Software Verification and Validation Plan (SVVP)
- Verification and Validation Reports (VVR)
- User documentation
- Open Item process
- Software Library
- Cyber Security

4.2.1 Software Requirement Specification (SRS)

The SRS shall be written by the Software Development Group and shall satisfy the requirements of IEEE Std. 730-2002, ~~/20/19/~~ and IEEE Std. 830-1998, ~~/22/21/~~, as endorsed by NRC Regulatory Guide 1.172, ~~/29/28/~~. This document shall be reviewed for adequacy in meeting those standards by an independent reviewer and then for traceability of requirements as per UFTR “Requirements Traceability Matrix (RTM)”, ~~/15/~~. The SRS shall be approved by the Software Development Lead.

4.2.2 Software Design Description (SDD)

The TXS Application Software SDD shall be written by the Software Development Group and shall meet the intent of IEEE Std. 1016-1998, ~~/23/22/~~. The SDD is a written representation of the TXS RPS application software, utilizing the FunBase tool, created to facilitate analysis, planning, and implementation in SPACE.

The TXS Application Software SDD is comprised of various views of the application software, including an overview of the system architecture, and a top-

UF/NRE UFTR	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 16 of 40</i>

level presentation of the important system functions. The SDD includes a library of all the standard SPACE design entities used in the software design. The SDD lists the important entities in the design, including the functional logic modules and their input and output modules. Other views include database tables listing the changeable parameters and information signals, together with some of their attributes, and communication interfaces.

This document shall be reviewed for adequacy in meeting those recommendations by an independent reviewer and then ~~for traceability of requirements from the SRS as per the UFTR RTM, /15/~~. The SDD shall be approved by the Software Development Lead.

4.2.3 Application Software Code Document

The TXS Application Software Code Document shall be written by the Software Development Group. The Code Document is a graphical representation of the TXS RPS Application Software developed using the TXS engineering tool SPACE.

The Application Software Code document provides the complete set of software function diagrams for the TXS project. The function diagrams are produced with the TXS engineering tool SPACE. The application code is automatically generated from the function diagrams documented here by using the qualified SPACE code generators. The document is based on the SDD, which implements the system requirements and function requirements, as well as the results of the basic design phase – specifically system architecture and ID coding concept.

This document shall be reviewed for adequacy by an independent reviewer and then ~~for traceability of requirements from the TXS Application Software SRS and SDD as per UFTR RTM, /15/~~. The Code Document shall be approved by the Software Development Lead.

4.2.4 Verification and Validation Plan

Project specific V&V Plans shall be generated following the guidance in the UFTR SVVP, /5/. These plans shall be generated by a member of the IV&V Group. These documents shall be reviewed for adequacy by another member of the IV&V Group in meeting the recommendations of IEEE Std. 1012-1998, ~~/26/25/~~, and IEEE Std. 1028-1997, ~~/24/23/~~, as endorsed by NRC Regulatory Guide 1.168, ~~/27/26/~~, and as applied to the software life cycle at the UFTR. The V&V Plans shall be approved by the IV&V Lead.

4.2.5 Verification Results Report and Validation Results Report

These Results Reports shall be generated following the guidance in the generic and project specific VVPs. They shall be prepared by a member of the

<i>UF/NRE UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 17 of 40</i>

IV&V Group and reviewed by another member of the group. The V&V Reports shall be approved by the IV&V Lead.

4.2.6 User Documentation

User documentation shall be provided to the customer in the form of the software documentation (SRS, SDD, Application Code, and any other related documentation) and the User Manual. User documentation shall be prepared by the Software Development Group and reviewed by an independent reviewer. The User Manual is jointly prepared by the Hardware & Testing and Software Development Groups. The preparation of the User Manual shall follow the guidance of IEEE Std. 1063-2001, ~~/25/24/~~.

4.2.7 Software Configuration Management Plan (SCMP)

The Application Software Configuration Management activities are defined, implemented, and managed in accordance with the UFTR “Software Configuration Management Plan (SCMP),” /4/.

The SCMP provides the method and tools to identify and control the TXS Application Software developed for a project.

Configuration control activities request, evaluate, approve or disapprove, and implement changes to the Application Software. Changes encompass both error correction and enhancement.

Schedule reporting tracks the completion of the Application Software throughout the project including additional activities documented to make changes.

The SCMP shall meet the requirements of IEEE 828-1990, ~~/21/20/~~, as endorsed by RG 1.169, ~~/28/27/~~. The SCMP shall be approved by the Software Development Lead.

4.3 Other Documentation

UFTR QAPP, /3/, outlines the phases for the project. Each Phase is described including typical phase input, tasks, processes, and outputs and results.

UFTR SSP, /6/, defines the software safety analysis activities to be performed to ensure that safety system software development achieves high functional reliability and design quality and is consistent with the defined system safety analysis requirements.

UFTR “Cyber-Security,” /14/, defines the administrative controls and design feature requirements for maintaining cyber-security for the project. The controls are designed to ensure a secure infrastructure for the project and to achieve the highest level of system integrity and protection from cyber-attack throughout the software development life cycle.

UFTR “Software Operations and Maintenance Plan,” /9/, describes the activities and resources that enable the UFTR to support and maintain TXS safety software after the software has been installed and is operational. This plan describes the process

UF/NRE UFTR	Prepared by		Reviewed by		QA-1, UFTR-QA1-01	
	Name:		Name:		Revision 1	Copy 1
	Date :	Initials:	Date :	Initials:	Vol. 1	Page 18 of 40

controls for responding to changes in customer requirements, offering upgrades, and reporting and correcting any defects or anomalies discovered in the software.

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 19 of 40</i>

5. Standards, Practices, Conventions, and Metrics

The software design process shall follow the standard TXS software lifecycle phases defined in the UFTR QAPP, /3/. The software design shall adhere to the processes defined in the UFTR SCMP, /4/, the UFTR SSP, /6/, and the UFTR “Software Operations and Maintenance Plan,” /9/. The UFTR SVVP, /5/, shall be used to verify and validate the software products.

Monitoring of compliance and adherence to these Operating Instructions shall be ensured through the Reviews and Audits defined in Section 6 of this Plan.

5.1 Documentation Standards

The documentation standards listed in the UFTR QAPP, /3/, shall be followed in developing the software documentation for the project.

5.2 Design Standards

The SRS shall be documented in a way which enables unique identification of ID-Codes for transmitters, signals, actuators, and annunciators (UFTR “ID Coding Concept,” /13/).

In the TXS Application Software SDD the safety functions, conditioning and online validation of input signals, actuation, and annunciation shall be documented in uniquely identified modules. Each module shall be described separately, with all interfaces and connections to other modules; therefore, each module can be tested and changed independently from other modules.

The logic structure of the TXS Application Software SDD in general and the module descriptions follow the structure of the TXS FunBase database. The Funbase tool offers a graphical user-interface, in which the functions can be designed according to the specification in the SRS. The logic structure of the SDD description in general and the module description in the SDD detail shall follow the structure of the used FunBase database.

The SPACE function diagrams shall have the same logical structure as the diagrams shown in the TXS Application Software SDD. They shall contain information about setpoints, time-delays, and the complete information about the interface as described in the TXS Application Software SDD. For each module a SPACE function diagram shall be specified. The SPACE tool creates a project database, in which all information about the safety functions, the function-specific setpoints, inputs and outputs is stored. The resulting function diagrams are a graphical interpretation of the safety system. The SPACE database is the basis for the automatic code-generation.

5.3 Coding Standards

The coding standards for the SRS are based on the description of functions in the FRS. Codes for signals, transmitters and actuators are described in interface lists and the UFTR “ID Coding Concept,” /13/.

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 20 of 40</i>

The TXS Application Software SDD is documented in a way which enables 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, with all interfaces and connections to other modules; therefore each module can be tested and changed independently from other modules.

The source code for the safety function is created by the TXS code generators. These are qualified TXS tools, which generate one source code file for each specified function diagram. Each source code file contains information about the respective function diagram, date and time of creation. The created source code files are not to be modified manually. They are subject to the rules described in the UFTR SCMP, /4/. The source code for the safety function is created by the TXS code generators.

For the TXS GSM and QDS Application Software, the code is generated using a combination of manually created scripts and Dialogue Screens, which are created by the Qt Designer tool utilizing special GSM and QDS libraries.

5.4 Commentary Standards

The SDD addressed in Section 4.2.2 contains commentary in the form of descriptions of the software functions developed using the TXS SPACE tools. Because the actual software code is generated using the qualified TXS code generators, as described in Section 5.3, it is not required to provide commentary in the actual generated code. The TXS SPACE tools use qualified function blocks for code development. These function blocks and their functionality are sufficiently described in the TXS function block manual.

5.5 Testing Standards and Practices

Testing Standards and practices for all the TXS projects shall be defined in the UFTR SVVP, /5/.

5.6 Software Quality Assurance Product and Process Metrics

Software quality metrics shall be used throughout the Software Life Cycle to assess the effectiveness of the SQAP. Software and design errors shall be recorded as “Open Items” during each phase of development. These items shall be tracked and trended to determine the progress in eliminating the errors present in the software and design.

Software metrics are developed from Open Item and Open Item Report Form to locate error prone areas of the code, design weaknesses, and testing flaws, etc. These reports include data that is used for trending which may include:

- Source of the defect (coding error, requirements deficiency, etc.)
- Type of defect or category
- Phase
- Quantity
- Priority

UF/NRE UFTR	Prepared by		Reviewed by		QA-1, UFTR-QA1-01	
	Name:		Name:		Revision 1	Copy 1
	Date :	Initials:	Date :	Initials:	Vol. 1	Page 21 of 40

- Method by which the defect was found
- Time of closure

Metrics are compiled by the IV&V group and are reported in the software V&V reports for analysis per the UFTR SVVP, /5/.

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 22 of 40</i>

6. Reviews and Audits

6.1 Reviews

Software reviews are conducted in accordance with IEEE Std. 730-2002, ~~/20/19/~~, and IEEE Std. 1028-1997, ~~/24/23/~~. 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. Software reviews are planned, performed, and documented in accordance with the UFTR "Software Reviews and Audits," /11/.

A minimum set of reviews is conducted as described in Sections 6.1.1-6.1.7.

6.1.1 Software Specification Review (SSR)

The SSR takes place during the Basic Design Phase after SRS is completed. The SSR may be performed along with the Preliminary Design Review. The SSR shall be prepared to ensure that the SRS adequately, technically, feasibly and completely reflects the design demands of the FRS, /12/. The SSR shall verify that the SRS was created in accordance with the standards listed in Section 4.2.1 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 the Configuration Control Board (CCB) in accordance with the UFTR "Software Reviews and Audits," /11/.

6.1.2 Preliminary Design Review (credited for the Architecture Design Review)

The Preliminary Design Review takes place during the Basic Design Phase and is conducted by the CCB. 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 activity shall be accomplished by the in accordance with the UFTR "Software Reviews and Audits," /11/.

6.1.3 Detailed Design Review

A Detailed Design Review shall be performed to verify that the detailed design, i.e., the SDD and the SPACE function diagrams, satisfies the requirements of the SRS and satisfies all functions specified in the FRS. 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.

<i>UF/NRE UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 23 of 40</i>

This activity shall be accomplished by in accordance with the UFTR “Software Reviews and Audits,” /11/.

6.1.4 Verification and Validation Plan Review (SVVPR)

A SVVR shall be performed after the initial SVVP is produced. The SVVPR shall evaluate the adequacy and completeness of the V&V methods defined in the SVVP. This activity shall be accomplished by the Project Management in accordance with the UFTR “Software Reviews and Audits,” /11/.

6.1.5 Managerial Reviews

Managerial Reviews may be held periodically by the Project Manager and/or Project Coordinator throughout the design and test processes, to assess the execution of the quality requirements in the contract specifications in accordance with the UFTR “Software Reviews and Audits,” /11/.

6.1.6 Software Configuration Management Plan Review

The SCMP 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 SCMP.

This activity shall be accomplished by the Project Manager in accordance with the UFTR “Software Reviews and Audits,” /11/.

6.1.7 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 the Project Manager in accordance with the UFTR “Software Reviews and Audits,” /11/.

6.2 Audits

Software Audits are conducted throughout the software life cycle and provide an independent evaluation of conformance of the software products and processes to applicable regulations, standards, and procedures, compliance with this Plan, IEEE Std 730-2002, ~~/20/19/~~, and IEEE Std. 1028-1997, ~~/24/23/~~. These audits are the responsibility of an independent QA Auditor and may include technical resources such as IV&V personnel, as necessary. The audits are planned, performed, and documented in accordance with the UFTR “Software Reviews and Audits,” /11/.

6.2.1 In-Process Audits

The reviews, inspections and requirements tracing activities described in the UFTR SVVP, /5/, are credited for satisfying the in-process audit requirements of IEEE Std 730-2002, ~~/20/19/~~. These independent verification inspections and

<i>UF/NRE UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 24 of 40</i>

reviews are performed by IV&V on software development products, including the FRS, SRS, SDD, test plans, specifications, cases, procedures, and results. Also included are code reviews for software not generated by SPACE. Deviations or discrepancies are recorded as Open Items. These reviews and inspections are held during the design phase and verify the consistency of the design, including:

- a. Code versus design documentation
- b. Interface specifications (hardware and software)
- c. Design implementations versus functional requirements
- d. Functional requirements versus test descriptions
- e. A review of the meeting minutes of the reviews specified in Section 6.1 to ensure that all findings have been incorporated and completed.

6.2.2 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. This Audit, along with the Functional Audit serves as a Configuration Audit per IEEE-828, ~~/21/20/~~. 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 SCMP. Found discrepancies shall be reported in the form of "Open Items" and are subject to the problem reporting process outlined in Section 8. This Audit shall be accomplished by the QA Project Auditor, in accordance with the UFTR "Software Reviews and Audits," /11/.

6.2.3 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 serves as a Configuration Audit per IEEE-828, ~~/21/20/~~. Audit shall be accomplished by the QA Project Auditor, in accordance with UFTR "Software Reviews and Audits," /11/.

6.2.4 Other Reviews and Audits - Software Process Audits

Software Process audits are conducted annually and are held to verify that this Plan, the SVVP, /5/, and the SCMP, /4/, are being complied with. Audit shall be accomplished by the QA Project Auditor, in accordance with the UFTR "Software Reviews and Audits," /11/.

UF/NRE UFTR	Prepared by		Reviewed by		QA-1, UFTR-QA1-01	
	Name:		Name:		Revision 1	Copy 1
	Date :	Initials:	Date :	Initials:	Vol. 1	Page 25 of 40

7. Testing

There are two types of testing utilized in the TXS software development process:

1. Simulation Testing using the SIVAT tool to validate the logic diagrams (UFTR “Software Test Plan – SIVAT Plan,” /7/)
2. Factory Acceptance Testing (FAT) (UFTR “Factory Acceptance Plan – FAT Plan,” /8/)

Project specific testing requirements shall be accomplished in accordance with the SVVP, /5/.

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 26 of 40</i>

8. Problem Reporting and Corrective Action

The Open Item Report is used to report defects in software or software documentation, as well as V&V documentation as specified in the SVVP, /5/. It is also used to track and close design and project issues. The open item report provides for recording the disposition of the defect and its resolution. Software changes resulting from error corrections are managed in accordance with the UFTR SCMP, /4/. Conditions adverse to quality, programmatic deficiencies, and errors found during the acceptance testing or the operational phase are documented within the UFTR corrective action procedure in accordance with the UFTR “Conduct of Quality Assurance,” /2/.

UF/NRE UFTR	Prepared by		Reviewed by		QA-1, UFTR-QA1-01	
	Name:		Name:		Revision 1	Copy 1
	Date :	Initials:	Date :	Initials:	Vol. 1	Page 27 of 40

9. Tools, Techniques, and Methodologies

This section identifies the software tools, techniques, and methods used to support the Software Quality Assurance processes.

9.1 Methodology for Generating the SRS

The method used by the Software Development Group for creating the SRS shall be used to follow the guidance in IEEE Std. 830-1998, ~~/22/21/~~ as endorsed by RG 1.172, ~~/29/28/~~. The Software Development Group shall assure that all of the specification requirements, functional requirements and software requirements are incorporated into the software design. ~~The software design group shall trace the requirements from the FRS into the SRS in accordance with the UFTR RTM, /15/.~~

9.2 Tools for Generating the SDD

The SDD shall be created by the Software Development Group using the tool FunBase which is a database management tool designed to facilitate organization of the Application Software functions and their respective internal and external Input/Output signals. ~~The software design group shall trace requirements from the SRS into the SDD in accordance with the UFTR RTM, /15/.~~

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

The tools for taking the function diagrams and converting them into software code are contained in the SPACE engineering system, which includes the source code-generators (Function Diagram Group Module and Run Time Environment) and the software for compiling, linking and locating. These tools are part of the qualified TXS system platform software package. The logic diagrams in the SDD shall be entered into the SPACE tool, and then the code shall be generated.

For the TXS, GSM and QDS Application Software, the code is generated using a combination of manually created scripts and Dialogue Screens, which are created by the Qt Designer tool utilizing special GSM and QDS libraries.

9.4 Tools for Software Simulation Testing

The tool for software simulation and validation testing is the TXS SImulation and VAlidation Tool (SIVAT). Validation testing in a summation environment can be one of the layers of validation testing that is used to ensure Application Software quality. In addition to integration and functional testing, the simulation tool may be used to perform debugging of the Application Software.

9.5 ~~Tools for Requirements Traceability Matrix (RTM)~~

~~Requirements tracing can be done using a matrix in worksheet form or by using a formal requirements management tool. A specific tool is not required, but the selected tool shall be approved for use by the IV&V Lead.~~

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 28 of 40</i>

9.69.5 Tools for Verification and Validation

Tools for V&V include:

- ~~The software RTM, used to trace requirements from the specification and the FRS to the SRS to the SDD to the code.~~
- The TXS software tools package (e.g. SIVAT, *reflist*, *hwparams*, *swparams*, *netload*, etc.), used to validate and document the software code
- The test environment of the field test equipment, including the TXS Test Machine, used to test the software implementation onto the system

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 29 of 40</i>

10. Media Control

The media for storing each deliverable work product and associated documentation are defined in the following subsections, together with a description of the safeguards to protect them from unauthorized access and inadvertent damage.

10.1 Media Control for the SDD

The database for the SDD (created by the tool FunBase) shall be stored on a network disk, accessible only to the Software Development Group. A procedure for a daily backup of the data of this disk is in use. Only the team members involved in the design process can access the database.

10.2 Media Control for Application Software

Application software is controlled in accordance with the UFTR "Software Library and Control," ~~46/15/~~. The SPACE project database is stored on a network server. Only the UFTR Project Manager and Project Coordinator or their designee involved in the specification process can access the project database. The organization of data backup procedures, schedules, and the storage of data saved on removable media is the responsibility of the Software Development Lead.

10.3 Code Control of TXS System Software

The TXS system software is a ready-made software-product, which is not to be modified during any phase of the Software Life Cycle. The verification of the identity of the system software is part of the SCM and has to be performed before each installation. Code Control of the TXS System Software is controlled in accordance with the UFTR SCMP, /4/.

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 30 of 40</i>

11. Supplier Control

AREVA NP Inc is the supplier of the TXS system software developed by AREVA NP GmbH, which has implemented an approved Software Quality Assurance program for the life cycle of the TXS software. The TXS system platform software is purchased for the UFTR digital control upgrade as a Safety-Related qualified product. Each item of software purchased is delivered with a Certificate of Conformance (CoC). Each CoC is reviewed to be current and applicable for its intended purpose.

All software in the TXS system software package shall be uniquely identified and shall be subjected to a receipt inspection in accordance with the UFTR SCMP, /4/.

In the software life cycle of the TXS projects at the UFTR, no other RPS-related software is required to be procured.

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 31 of 40</i>

12. Records Collection, Maintenance, and Retention

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

All documents produced during the project in accordance with the UFTR QAPP shall be stored in the UFTR project dedicated server. This includes, but is not limited to:

- Software Requirements Specification (SRS)
- Software Design Description (SDD)
- Code Documents (SPACE Listings, Code Configurations, List of Changeable Parameters, etc.)
- Verification and Validation (V&V) Reports
- All test-procedures and test-results of the SIVAT tests
- All test-procedures and test-results of the test field tests
- Review Reports
- Audit Reports

Retention periods for all QA records are specified in the UFTR QAPP, /3/.

<i>UF/NRE UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 32 of 40</i>

13. Training

All Design and IV&V personnel shall be trained in accordance with the UFTR “Software Training Plan,” /10/, and on the provisions of the UFTR QAPP, /3/, and the other UFTR plans and procedures as necessary to implement this Plan.

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 33 of 40</i>

14. Risk Management

Because of the safety features of the UFTR, the risk management is not applicable.

UF/NRE UFTR	Prepared by		Reviewed by		QA-1, UFTR-QA1-01	
	Name:		Name:		Revision 1	Copy 1
	Date :	Initials:	Date :	Initials:	Vol. 1	Page 34 of 40

15. Glossary

15.1 Definitions

Anomaly, [IEEE Std. 1012-1998, #26#25/]:

Any condition that deviates from the expected based on requirements, specification, design, documents, user documents, standards, etc., 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 TXS Instrumentation and Control (I&C) system. It is documented and generated by the Engineering SPACE Tool. The platform system software uses this configuration data in order to carry out the application specific functionality of the I&C system.

Baseline, [IEEE 610.12-1990, #9#18/]:

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 610.12-1990, #9#18/]:

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.

Code, [IEEE 610.12-1990, #9#18/]:

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

Component, [IEEE 610.12-1990, #9#18/]:

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

Configuration, [IEEE 610.12-1990, #9#18/]:

- 1) The arrangement of a computer system or component as defined by the number, nature, and interconnections of its constituent parts.
- 2) 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 610.12-1990, #9#18/]:

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 35 of 40</i>

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 Identification, [IEEE 610.12-1990, #9#18/]:

- 1) An element of configuration management, consisting of selecting the configuration items for a system and recording their functional and physical characteristics in technical documentation.
- 2) The current approved technical documentation for a configuration item as set forth in specifications, drawings, associated lists, and documents referenced therein.

Configuration Item, [IEEE 610.12-1990, #9#18/]:

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 610.12-1990, #9#18/]:

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.

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 FRS or the SRS. These discrepancies are initially documented on the open item list and are evaluated for further action.

FunBase

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

<i>UF/NRE</i> <i>UFTR</i>	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 36 of 40</i>

Functional Requirements Specification

A document provided by the customer or his agent that describes in detail the functions of the system to be installed new or replaced. The FRS will include both hardware and software functions of the system. This document can be called by a different name, but whatever document is provided by the customer to meet this function will fit this definition.

Interface, [IEEE 610.12-1990, #18/18/]:

- 1) A shared boundary across which information is passed. This boundary includes design interfaces between design organizations.
- 2) A hardware or software component that connects two or more other components for the purpose of passing information from one to the other.
- 3) To connect two or more components for the purpose of passing information from one to the other.
- 4) To serve as a connecting or connected component as in 2).

Interface Control, [IEEE 610.12-1990, #18/18/]:

In configuration management, the process of:

- identifying all 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

Open Item

Any item which constitutes an error or anomaly from the required status or condition of a properly completed project. Open Items are each given a record in a database with a unique (to the project and Unit) identifier.

The entry contains information to track the cycle of the item from initiation to final resolution.

Physical Configuration Audit, [IEEE 610.12-1990, #18/18/]:

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

SIVAT

SIVAT (Simulation & Validation Tool) allows the functionality of the I&C system engineered in SPACE to be tested by simulation. Simulation is based on the code generated by the function diagram group code generator and the Runtime Environment code generator. This enables engineering errors to be detected at an early stage. The objective of the test is to verify that the requirements have been translated into function diagrams without errors, and that the software automatically generated from

UF/NRE UFTR	<i>Prepared by</i>		<i>Reviewed by</i>		<i>QA-1, UFTR-QA1-01</i>	
	<i>Name:</i>		<i>Name:</i>		<i>Revision 1</i>	<i>Copy 1</i>
	<i>Date :</i>	<i>Initials:</i>	<i>Date :</i>	<i>Initials:</i>	<i>Vol. 1</i>	<i>Page 37 of 40</i>

these functions diagrams provides the functionality required in terms of input and output response. The tests cover interface to the Runtime Environment, use of correct function blocks and whether they have been correctly connected and parameterized. The failure of Input/Output modules, processing modules and data messages can be simulated. The tests are run using scripts that define the input signals of the I&C system and the simulation run. The test results are recorded in log files and plots for further evaluation. Process models can also be linked into the simulator to perform closed-loop tests.

Software, [IEEE 610.12-1990, #9#18/]:

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

Software Design Description, [IEEE 610.12-1990, #9#18/]:

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 Library, [IEEE 610.12-1990, #9#18/]:

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 610.12-1990, #9#18/]:

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

Software Life Cycle Phases

The Application Software life cycle phases for the UFTR are Basic Design, Detailed Design, Testing, Installation, Commissioning, and Final Documentation

Software Simulation Testing

Using the SIVAT tool to functionally test software modules generated by the SPACE tool.

SPACE

SPecification And Coding Environment (SPACE) system comprises the tools used for engineering and maintenance of the TXS I&C software. Engineering in this context 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 (*reflist* and *scanmic*)

UF/NRE UFTR	Prepared by		Reviewed by		QA-1, UFTR-QA1-01	
	Name:		Name:		Revision 1	Copy 1
	Date :	Initials:	Date :	Initials:	Vol. 1	Page 38 of 40

- Software Loading
- Load Analysis tool
- Database administration

System Software, [IEEE 610.12-1990, #9#18/]:

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

Test Plan

A document describing the approach to be taken for intended testing activities. It identifies the items to be tested, the testing to be performed, test sequences, personnel requirements, and evaluation criteria.

TXS Project Phases

The TXS Project phases are Project Start-Up/Conceptual Engineering, Basic Design, Detailed Design, Manufacturing, Testing, Installation & Commissioning, and Final Documentation.

TXS Project Basic Design Phase

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

TXS Project Detailed Design Phase

The activities that result in a completely specified and SIVAT tested I&C system which fulfills all requirements specified in the contract.

TXS Test Phase

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

Unit, [IEEE 610.12-1990, #9#18/]:

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.

Version, [IEEE 610.12-1990, #9#18/]:

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

UF/NRE UFTR	Prepared by		Reviewed by		QA-1, UFTR-QA1-01	
	Name:		Name:		Revision 1	Copy 1
	Date :	Initials:	Date :	Initials:	Vol. 1	Page 39 of 40

Verification and Validation, [IEEE 610.12-1990, A9/18/]:

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.

15.2 Abbreviations and Acronyms

FAT	Factory Acceptance Test
FD	Function Diagram
FDG	Function Diagram Group
FRS	Functional Requirements Specification
GSM	Graphic Service Monitor
I&C	Instrumentation and Control
ID	Identification
IV&V	Independent Verification and Validation
IEEE	Institute of Electrical and Electronic Engineers
I/O	Input/Output
QDS	Qualified Display System
RG	Regulatory Guide
RTM	Requirements Traceability Matrix
SCMP	Software Configuration Management Plan
SCR	Source Code Review
SDD	Software Design Description
SDR	Software Design Review
SIVAT	Simulation Based Validation Tool
SPACE	Specification and Coding Environment
SQA	Software Quality Assurance
SRS	Software Requirements Specification
SSR	Software Specification Review
Std	Standard
SVV	Software Verification and Validation
SVVP	Software Verification and Validation Plan
SVVPR	SVVP Review
SVVR	Software Verification and Validation Report
TXS	TELEPERM XS
V&V	Verification & Validation

UF/NRE UFTR	Prepared by		Reviewed by		QA-1, UFTR-QA1-01	
	Name:		Name:		Revision 1	Copy 1
	Date :	Initials:	Date :	Initials:	Vol. 1	Page 40 of 40

16. SQAP Change Procedure and History

The history of changes for this SQAP is documented in the Record of Revisions Section.

No change made to this plan shall change the provisions in the UFTR QAPP, /3/, without the QAPP being changed and approved beforehand.