

AREVA NP RESPONSE to FOURTH REQUEST FOR ADDITIONAL INFORMATION (RAI)

ANP-10272, "SOFTWARE PROGRAM MANUAL FOR TELEPERM XS™ SAFETY SYSTEMS
TOPICAL REPORT" (TAC NO. MD3971) PROJECT NUMBER 733

RAI-71 Question: What is AREVA NP's justification for deviating from IEEE 1012-1998 in allowing the software design organization to design and run verification and validation (V&V) tests in place of the V&V organization?

NRC Concern: The V&V organization exists as a separate check and balance to the design engineering and software organizations to provide a high degree of confidence in the compliance with functional, performance, and interface requirements; and in the completeness and correctness of the software or system in question. The V&V team, in some ways, may be viewed as a layer of diversity and defense-in-depth in the quality, design, and testing processes.

IEEE 1012 makes the generation and execution of various test plans the duty of the V&V organization.

Allowing the software design organization to design and run V&V tests in place of the V&V organization indicates a loss of that diverse defensive layer in a safety-critical stage of software testing. Various errors of interpretation and implementation which may have been introduced into the software by the thinking of the software design organization will likely be masked in the design and execution of the V&V tests through those same pathways of interpretation if executed by the same organization. AREVA NP has not explained or justified how its procedures and methods will mitigate these possible errors in lieu of fully complying with IEEE 1012 (by maintaining the V&V organization's complete responsibility for the design and execution of V&V tests). It has also not detailed any specific requirements of the V&V organization in this scenario.

It should be noted, however, that while the responsibility of generating V&V test plans and procedures lies solely with the V&V organization, the design engineering personnel may carry out the actual tests.

The staff requires additional information to determine if the proposed alternative to the requirements of IEEE 1012 will provide an equivalent confidence in a high quality test process, and therefore an equivalent confidence in the safety of the resultant system.

AREVA NP's response should include the following:

- Documentation of independent V&V Group's Assessment of Testing
- Documentation of V&V Group's Role/Interaction with design and the Test Group
- Documentation of how Problems identified by the test are resolved and the V&V Group's role in that process
- Clear delineation of responsibility and authority of the V&V team over the V&V testing, planning, design, execution, and review

Applicant Reference(s):

ANP-10272

Simulation Testing (Page 6-7)

As a minimum, the verification and validation engineer reviews the simulation test plan and results of the testing to ensure that the requirements are adequately tested.

1. *In the event that SIVAT testing is only performed by design engineering with a complete factory acceptance test, **the verification and validation team only performs the reviews.***

AREVA NP Response to RAI 71: The following response addresses validation testing in the context of the IEEE Std 1012-1998, "IEEE Standard for Software Verification and Validation," using the TELEPERM XS (TXS) technology.

AREVA NP has agreed to modify its approach to validation testing.

The Software Verification and Validation Plan (SVVP) conforms to the guidance of the applicable recommendations of IEEE Std 1012-1998, as endorsed by Regulatory Guide 1.168, Revision 1. 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 (V&V) group may get assistance from the Software Design group, Hardware Design group, and the Test group for the preparation of validation test specifications, procedures, and reports and the performance of test tasks; however, these support personnel shall work under the supervision of the V&V group. These support personnel may not develop 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, including the Factory Acceptance Test (FAT). The test document workflow and control points are shown in Figure 71-1 to ensure that the independence of validation testing is maintained.

Verification and Validation Organization

The V&V group is responsible for independent V&V activities described in the SVVP. The V&V group performs independent V&V reviews and validation tests.

The V&V group is comprised of a diverse team of personnel from AREVA NP, Inc. (NL-A). Additional support can be provided by the specialized hardware design group of NL-G, and the specialized V&V group of Euriware, an AREVA subsidiary based in Europe. The V&V 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, Revision 1. V&V personnel may not participate in any design preparation activities; however, the V&V group can perform the Appendix B independent design review for the Software Design group, Hardware Design group, and the Test group, provided the requirements for Appendix B Design Control are

satisfied. The V&V group manager has a dotted line responsibility to the Quality Assurance (QA) group. From a project management perspective, the V&V group is considered part of the overall project team.

The V&V group has sufficient resources (i.e., budget and staff) and authority to ensure V&V activities are not adversely affected by commercial and schedule pressures.

Responsibilities for Testing

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

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

V&V personnel shall be trained in the use and output of SIVAT, for the preparation or verification of software test documents and validation testing. V&V personnel shall be familiar with acceptance test procedures, predicting test results, and the form of the generated outputs from the TXS System for validation testing.

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

Application Software Validation Testing

The SIVAT tool can be used to perform application software integration and functional testing (see response to Issue 5). Application software validation through SIVAT testing is one of the layers of validation testing that is used to ensure application software quality.

The TXS development process has features that are specifically designed to improve the reliability of the application software. The use of a standard Function Block library provides a large experience base for the standard modules. The use of the SPACE tool to generate code automatically eliminates an important human error source associated with manual code generation. Use of the SPACE tool eliminates both errors of translation and the introduction of complexity by engineers trying to optimize application coding. The SPACE tool and Function Block library are subject to the generic qualification process described in the TXS topical report. The generic qualification approach provides a very high degree of validation independence

commensurate with the importance of generic system qualification. The TXS development process requires the use of the SPACE tool and Function Block library to create code from the Function Diagrams and Function Diagram Group modules. The generic qualification process and the use of the SPACE tool provide the foundation for project-specific application software validation testing.

SIVAT testing, when used for validation testing to satisfy IEEE Std 1012-1998 requirements, is performed under the direction of the V&V group. SIVAT is used to perform an additional layer of application software validation testing that falls between Function Block testing (component testing) and the FAT, which serves as the system integration and acceptance testing. The SIVAT test plans, specification procedures, and reports are prepared in accordance with the SVVP and 10 CFR Part 50 Appendix B QA requirements.

The SIVAT tool is used to validate the application software functionality. It can also simulate certain TXS malfunctions (i.e., failure of an input/output module, failure of a message, and failure of a complete TXS central processor unit) to verify that the response to these faults is as intended. SIVAT enables the V&V engineer to compare the validation results to the software requirements specifications. The V&V group uses the software requirements traceability matrix to ensure that software requirements have been tested.

The benefit of application software validation testing with SIVAT 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 quality assurance observation and monitoring) and performing application software validation testing with SIVAT earlier in the process. IEEE Std 1008-1987, "IEEE Standard for Software Unit Testing," 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 with SIVAT serves to reduce project risks earlier in the development process.

System and Acceptance Testing

The system test, including FAT, fulfills the requirement for system integration and acceptance validation testing. Additional application software integration and functional test cases are added to the scope of system testing for the case where SIVAT testing is not used to satisfy IEEE Std 1012-1998 validation requirements for application software validation testing. The FAT is a formal project milestone that is attended by both AREVA NP Quality Assurance and customer personnel.

The generic TXS platform software and hardware integration is subject to the generic qualification process described in the TXS topical report. The generic qualification approach provides a very high degree of validation independence commensurate with the importance of

generic system qualification. The generic qualification work provides the foundation for the project-specific system testing, including FAT.

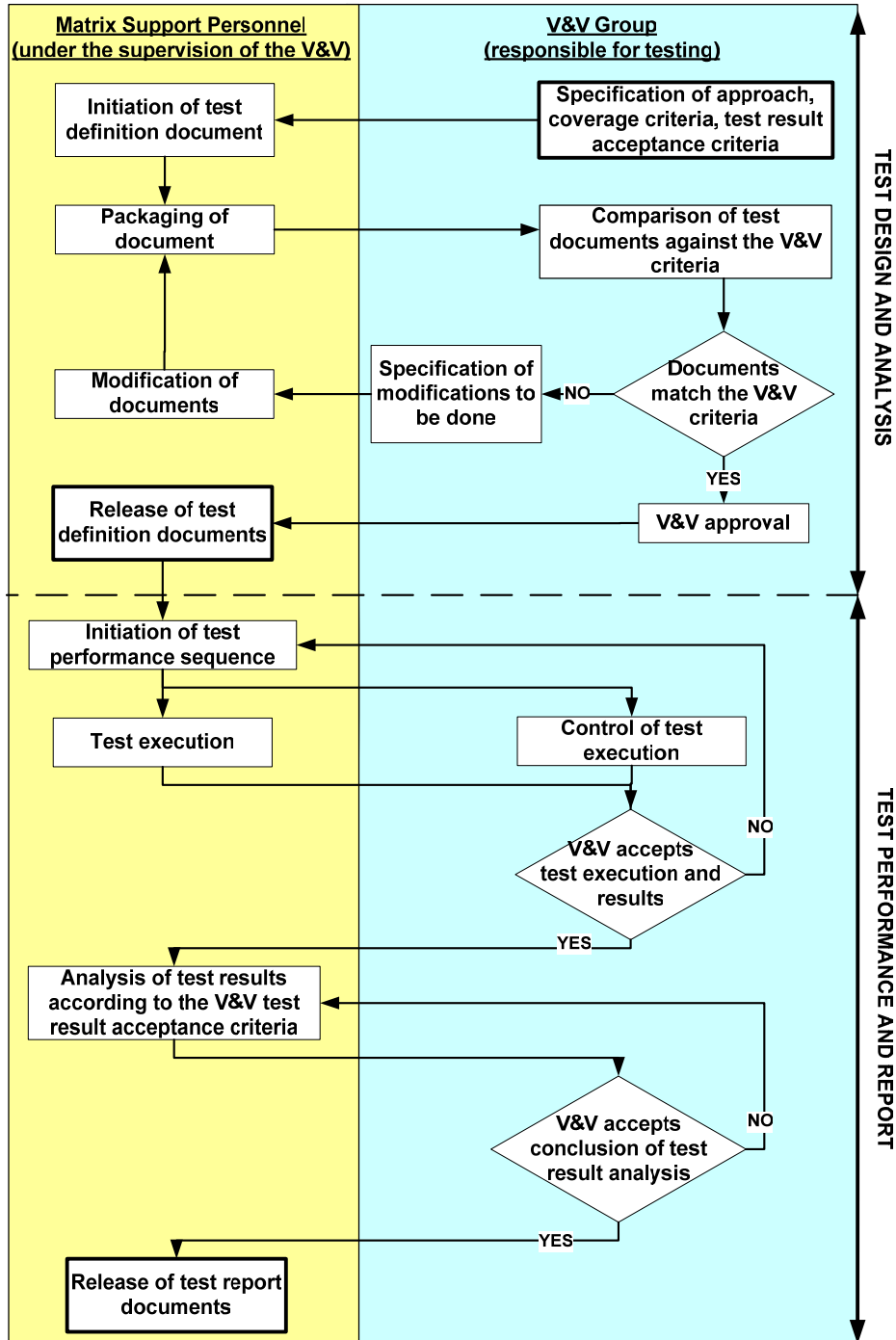
The system test, including FAT, is performed under the direction of the Verification and Validation group. The system test plans, specifications, procedures, and reports are prepared in accordance with the Software Verification and Validation Plan and 10 CFR Part 50 Appendix B QA requirements.

The Verification and Validation group uses the software requirements traceability matrix to ensure that system functional requirements have been tested in the system tests. Similarly, the Verification and Validation group uses the software requirements traceability matrix to ensure that software requirements have been tested if application software integration and functional testing is performed during system testing. The Verification and Validation group independently verifies that the software versions being tested match those listed in the Software Configuration Management Plan.

Software Program Manual Changes

The Software Program Manual will be revised to incorporate this information.

Figure 71-1 – Test Document Work Flow and Control Points



RAI-72 Question: Where the AREVA NP SPM states that it conforms, complies, or uses similar language with regard to a particular industry standard, NRC regulation, or NRC guidance, is the SPM in fact stating 100% compliance to that reference (except, of course, where specific exceptions are identified and described in the SPM)? This is a “yes” or “no” question. If the answer is “no,” identify deviations from the standards, regulations, and guidance used in the SPM.

NRC Concern: Within the text of the SPM, there are multiple instances of conformance claims. The majority of the conformance claims use similar language to describe conformance. Some sections of the SPM describe deviations from conformance where previous text in the same section seemed to describe full conformance. Other sections of the SPM state “applicable” conformance, but do not go on to list what is not applicable nor give explanation for that claim. This intermittent identification of deviations from standards, guidance, and regulations seems to indicate that AREVA NP only generally follows those documents in executing the SPM, and gives no guarantee that all intended exceptions have been identified in the SPM.

Without a specific list of exceptions in the SPM, the NRC considers that the SPM dictates full compliance with referenced standards, guidance, and regulations. The associated SE for the SPM will be written to reflect that understanding, and any combined license application (COLA), license amendment request (LAR), or design certification (DC) referencing the SPM will be held to the same scrutiny of full compliance unless those documents take specific exception. Furthermore, in the case of a review where it is found that an application referencing the AREVA NP SPM has deviated from the SPM without stating explicit exceptions, the application will have to be considered in non-compliance with the AREVA SPM.

Where the SPM states that it takes exception to compliance, or only applies to applicable sections, without listing and explaining those exceptions or applicability, AREVA should revise the SPM’s language and add amplifying information.

Applicant Reference(s):

The following is a list of examples, though it is not exhaustive:

ANP-10272

Test Planning (Page 3-5)

*Testing activities **follow** the guidance of IEEE 829 (Reference 19), which is endorsed by Regulatory Guide 1.170 (reference 7)...*

Methodology for Generating the Software Requirements Specification (Page 3-7)

*The software design group **follows** the guidance in IEEE 830 (Reference 20), which is endorsed by Regulatory Guide 1.172 (Reference 9), **as the preferred method** for the creation of the SRS...*

Software Safety Plan (Page 4-1)

The plan follows the concepts of IEEE 1228 but does not fully comply (Reference 28).....

Software Verification and Validation Plan (Page 6-1)

*The Software Verification and Validation Plan follows the guidance of the **applicable** recommendations of IEEE 1012, which is endorsed by Regulatory Guide 1.168. **One area of exception** with regard to the IEEE Standard 1012 is...*

Test Plan (Page 9-4)

*Software simulation testing with SIVAT is planned and executed in accordance with procedures **following the applicable** recommendations of IEEE 1008 (Reference 21), which is endorsed by Regulatory Guide 1.171 (reference 8).*

AREVA NP Response to RAI 72: The Software Program Manual will be revised to provide the following clarifications:

Section 3.1 - The Software Quality Assurance Plan fulfills the requirements for a software quality assurance plan and conforms to IEEE Std 730-2002. Some of the activities described in the Software Quality Assurance Plan are performed by the independent AREVA NP QA organization. The governing document for activities performed by the QA organization (i.e., reviews, surveillances, and audits) is the AREVA NP Quality Management Manual. This information is not duplicated in the Software Quality Assurance Plan; instead, it is simply referenced.

Section 3.4 - The design reviews and process audits conform to the guidance of IEEE Std 1028-1997, as endorsed by Regulatory Guide 1.168, Revision 1.

Section 3.5.1 - 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.

Section 3.8.1.1 - The Software Design group ensures that the SRS conforms to the guidance in IEEE Std 830-1993, as endorsed by Regulatory Guide 1.172.

Section 3.13.1 – The AREVA NP risk management process conforms to the guidance in IEEE Std 7-4.3.2-2003 clause 5.3.6, as endorsed by Regulatory Guide 1.152, Revision 2.

Section 4.2 – The organization approach for the Software Safety Plan used by AREVA NP meets the intent of IEEE Std 1228-1994. (See response to RAI 83 for additional information.)

Section 4.3 - When combined together these activities also satisfy the requirements for a software hazards analysis and meet the intent of IEEE Std 1228-1994. (See response to RAI 83 for additional information.)

Section 4.3.4 – The FMEA conforms to the guidance of IEEE Std 379-2000, as endorsed by Regulatory Guide 1.53, Revision 2.

Section 5.1.1 – The Software Configuration Management Plan conforms to the guidance of IEEE Std 828-1990 and IEEE Std 1042-1987, as endorsed in Regulatory Guide 1.169, with the exception of the use of a configuration control board.

Section 6.0 - 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, Revision 1. 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.

Section 7.0 - The Software Operations and Maintenance Plan conforms to the life cycle planning for operations and maintenance guidance of IEEE Std 1074-1995, as endorsed by Regulatory Guide 1.173.

Section 9.1.2 – The SRS conforms to of the guidance of IEEE Std 830-1993, as endorsed by Regulatory Guide 1.172.

Section 9.2.1 - Software testing follows written test plans that conform to the guidance IEEE Std 829-1983, as endorsed by Regulatory Guide 1.170.

Section 9.2.1 - Application software validation testing with SIVAT is planned and executed in accordance with procedures that conform to the applicable guidance of IEEE Std 1008-1987, as endorsed by Regulatory Guide 1.171.

Also, see responses to RAIs 71, 73, 79, 81, 83, and 84.

RAI-73 Question: What is AREVA NP's justification for crediting SIVAT testing in reducing the scope of the factory acceptance test (FAT) and other unit and integration testing?

NRC Concern: AREVA NP's SPM, in combination with the TELEPERM XS topical report, does not contain enough information for the NRC to make a determination as to the acceptability or safety of the software testing process in terms of the coverage of the testing or with regard to the tools used to accomplish that testing.

Specifically, the following information is lacking:

1. The scope and coverage of SIVAT and FAT testing, in combination and separately
2. The capabilities, qualification, and implementation of SIVAT
3. Information on integration and unit testing; and how that testing may be satisfied by a combination of SIVAT and FAT testing
4. Details of the SIVAT and FAT tests
5. Justification for reducing the FAT tests
6. Guidelines for determining the extent to which FAT coverage may be reduced by SIVAT testing

In general, there is no way for the NRC to understand what is meant by a reduction in FAT testing with the information currently provided by AREVA NP. It appears that SIVAT simulates the software's operation. With the use of SIVAT, AREVA NP needs to demonstrate how FAT will fully verify and validate software/hardware integration aspects. AREVA NP should provide procedures and processes that will ensure software/hardware integration aspects are appropriately addressed in the proposed V&V scheme.

It should also be understood that SIVAT is not an approved tool. While the concept of simulated testing was mentioned briefly in the TELEPERM XS topical report, the SIVAT tool was not described, nor was it approved by the associated SER. As such is the case, a reference to the TELEPERM XS topical or SER is not considered sufficient justification or explanation for the SIVAT, FAT, unit, and integration testing proposed by the SPM.

The SPM relies heavily on the concept of SIVAT as a tool, in combination with FAT, for V&V in place of more traditional methods. As such is the case, the approval of the SPM may be predicated upon NRC's understanding and acceptance of the SIVAT tool and AREVA NP's FAT methodologies and procedures.

AREVA NP Report No. NGLP/2004/en/0094, "TELEPERM XS Simulation - Concept of Validation and Verification" should be submitted to support AREVA NP's explanation of SIVAT.

Applicant Reference:

Simulation Testing (Pages 6-7,8)

*If the verification and validation team performs tracing or SIVAT testing and tracing, **the testing can be credited to reduce the scope of the factory acceptance test.** Three options can be used to determine the verification and validation scope:*

1. *In the event that SIVAT testing is only performed by design engineering with a complete factory acceptance test, the verification and validation team only performs the reviews.*
2. *The verification and validation team can trace the requirements through the SIVAT testing as performed by design engineering, **in which case the scope of the factory acceptance testing will be reduced.***
3. *The verification and validation team can plan and perform SIVAT testing in addition to tracing, in which case the factory acceptance test scope will be reduced.*

AREVA NP Response to RAI 73: ¹ The TXS system is a fully integrated suite of hardware and software designed specifically for nuclear safety applications. The TXS system has significant nuclear operating experience. The TXS platform has been fully qualified as an integrated platform. The TXS system is described in AREVA NP Topical Report EMF-2110(NP), "TELEPERM XS: A Digital Reactor Protection System," Revision 1 (referred to as the TXS Topical Report). ² NRC approved the TXS Topical Report in a safety evaluation report (SER) issued in May 2000. ³

The overall qualification process for the TXS system is shown in Figure 2.2 from the TXS Topical Report. 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. 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 TXS Topical Report.

The overall application independent qualification process is described in Section 2.2 of the TXS Topical Report. The TXS platform qualification process is shown below (reproduction of TXS Topical Report Figure 2.2).

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 TXS Topical Report describes part of the AREVA NP activities.

1 Much of the information in this response was previously provided to NRC in a letter from Ronnie L. Gardner (AREVA NP Inc.) to Document Control Desk (NRC), "Follow-up Actions from December 19, 2007, NRC Audit of ANP-10272, "Software Program Manual for TELEPERM XS Safety Systems Topical Report," Program Implementation (TAC No. MD3971)" NRC: 08:008, January 24, 2008.

2 Siemens Topical Report EMF-2110, Revision 1, "TELEPERM XS: A Digital Reactor Protection System," May 2000 Enclosure to letter, James F. Mallay (Siemens Power Corporation) to Document Control Desk (NRC), "Publication of EMF-2110(NP)(A) Revision 1, TELEPERM XS: A Digital Reactor Protection System," NRC:00:033, July 12, 2000.

3 Letter dated May 5, 2000, from Stuart A. Richards, NRC, to Jim Mallay, Siemens Power Corporation, "Acceptance for Referencing of Licensing Topical Report EMF-2110 (NP), Revision 1, "TELEPERM XS: A Digital Reactor Protection System" (TAC NO. MA1983)," and associated Safety Evaluation Report.

Section 2.4.3.3.3 of the TXS 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 TXS Topical Report. The TXS Software Program Manual was prepared to describe the specific system qualification process for TXS projects in the U.S.⁴

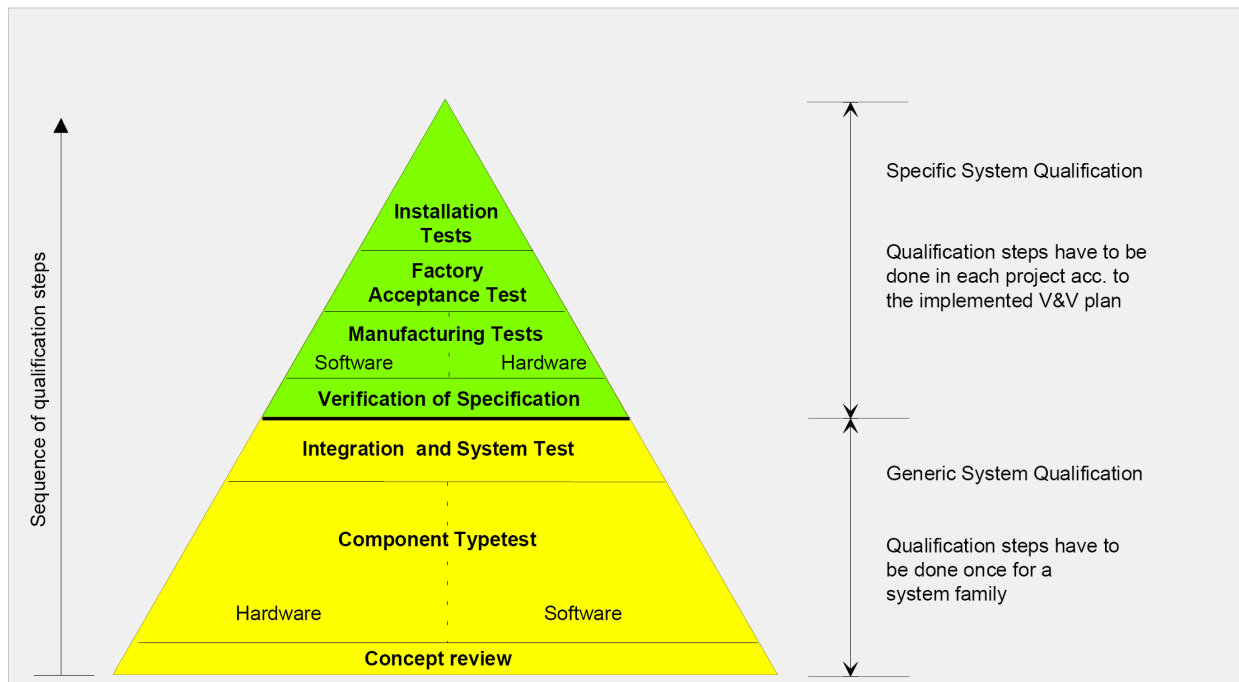


Figure 2.2 from TXS Topical Report

TXS Project-Specific Activities⁵

The TXS Topical Report described a general framework for the implementation of individual projects using the TXS technology in Section 5.1.3. An integral part of that framework is the use of the TXS engineering tools set for the development of application software. A key feature of the TXS tool set is the safety-related automatic code generator in the Specification and Coding

4 AREVA NP Topical Report ANP-10272, Revision 0, "Software Program Manual for TELEPERM XS Safety System Topical Report," December 2006, Enclosure to letter, Ronnie L. Gardner (AREVA NP Inc.) to Document Control Desk (NRC), Request for Review and Approval of ANP-10272, "Software Program Manual TELEPERM XS Tm Safety Systems Topical Report," NRC:06:061, December 21, 2006.

5 This information was previously provided during the NRC review of the TXS Software Program Manual in response to RAI 41 in a letter from Ronnie L. Gardner (AREVA NP Inc.) to Document Control Desk (NRC), "Response to Third Request for Additional Information Regarding ANP-10272, 'Software Program Manual for TELEPERM XS™ Safety Systems Topical Report' (TAC No. MD3971)," NRC: 07:056, on October 26, 2007.

Environment (SPACE) tool, which was approved as part of the TXS Topical Report. The development of TXS application software is predicated on the use of the SPACE tool for code generation.

The TXS Topical Report describes the simulator-based validation process for TXS application software in Section 2.4.3.3.2. The simulator-based validation tool described in the report is SIVAT. The role of the simulator-based validation tool in the standard AREVA NP engineering process for TXS project implementation is shown in TXS Topical Report Figure 2.8. The correctness of TXS code generation in the course of application projects is covered by validation activities (i.e., software validation testing with SIVAT or during system testing). RETRANS analysis was not considered to be part of the standard TXS 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 TXS Topical Report. The TXS Software Program Manual describes the specific system qualification process for TXS projects in the U.S. This process uses the SIVAT (Simulation-based Validation Tool), as described below.

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

Physical software integration occurs during the system test stage, when the application software is loaded on the TXS processors. The project-specific system test plan (including FAT) 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.

Verification of SPACE Tool Automatic Code Generator

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

ISTec (Institute of Safety Technology) is a subsidiary of the German Society for Reactor Safety (GRS). GRS-ISTec has a major role in the TXS 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 tool code generator complies with the design rules given for the generated Function Diagram and Function Diagram Group modules, and that the functions contained in the generated code are equivalent with the specification data contained in the SPACE database. GRS-ISTec is the RETRANS tool owner and responsible for tool support. In this context, RETRANS is used to validate correct performance of the automatic code generator as part of the generic qualification process.

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

RETRANS Tool Capabilities

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

The verification of source code generated by TXS code generator is performed by analysis and retranslation of TXS application code generated by SPACE tool code generator. The reconstructed specifications from the code retranslation are compared with specification data contained in the TXS engineering database (SPACE database). This feature was described in the TXS 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 (SER) for the TXS 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 used this feature for a period of time then replaced the capability with a separate tool (**rediff**). AREVA NP developed the **rediff** tool for ergonomic and efficiency reasons. The AREVA NP **rediff** tool is used for every TXS project to support TXS application software verification tasks. The **rediff** tool was designed and implemented using internal quality assurance procedures.

Experience with RETRANS

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

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

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

The RETRANS verification relies on the (generically defined) program structure of generated Function Diagram and Function Diagram Group modules and on the data model of the SPACE database. Once it has been verified and tested that a code generator version generates code in compliance with this program structure and equivalent with the input data from the SPACE database, RETRANS cannot identify errors in the course of a TXS application project implementation. Therefore, RETRANS analysis was not defined to be a step in the TXS 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 TXS 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 TXS systems into nuclear power plants. No findings concerning the SPACE code generator have been identified in later versions.

Changes to RETRANS Tool

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

RETRANS Tool	Runs with Operating System	For Use with TXS 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 (used for Ocone)
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 TXS application code. These structures have not changed since the initial generic qualification of TXS. Later developments of RETRANS consisted of adaptations to minor modifications in the SPACE data base structure and adaptations to new operating system versions of the engineering workstation. RETRANS is going to be adapted by GRS-ISTec to the current release of the TXS core software in 2008. GRS-ISTec plans to test every modification of the TXS code generator presented by AREVA NP using RETRANS.

Simulator-Based Validation of TXS Application Software

The TXS Topical Report describes the simulator-based validation process for TXS application software in Section 2.4.3.3.2. The simulator-based validation tool described in the report is SIVAT, which was being developed concurrently with the preparation of TXS Topical Report.⁶ The role of the simulation-based validation tool in the standard AREVA NP engineering process for TXS project implementation is shown in TXS Topical Report Figure 2.8. The correctness of TXS code generation in the course of application projects is covered by validation activities (i.e., software validation testing with SIVAT or during system testing).

The NRC SER for the TXS Topical Report stated in Section 4.4 that:

The IV&V processes address all phases of the Siemens software life cycle up to the testing of plant-specific applications. The staff did not address plant-specific applications of IV&V activities, as these activities were not in the scope of the staff review.

TXS Software Program Manual augments the generic system qualification process described in the TXS Topical Report with a standard engineering process to be used to develop TXS application software for U.S. projects.

6 During the December 19, 2007, meeting with NRC, AREVA NP supported its interpretation of the TXS Topical Report by using the original 1999 slides from the initial meeting with NRC on that subject. The slides from the TXS Engineering Process section showed that SIVAT was the validation tool specified for use in developing application software. Similarly, the slides from the section on SPACE Qualification Processes discussed the development of an independent code verification tool for checking code generator output (RETRANS) and the use of tool-based checks for generated TXS application software for first I&C projects.

A detailed description of the SIVAT and its application for validation testing of the TXS application software is provided in AREVA NP Report No. NGLP/2004/en/0094, "TELEPERM XS Simulation - Concept of Validation and Verification." The basis for its use is also addressed in the report. AREVA NP Report No. NGLP/2004/en/0094 is the latest version of the report. AREVA NP Report No. NGLP/2004/en/0094 has been provided to NRC in an April 3, 2008 letter from Duke Energy to NRC entitled "License Amendment Request for Reactor Protective System/Engineered Safeguards Protective System Digital Upgrade, Technical Specification Change (TSC) Number 2007-09, Supplement 1." It is listed as LAR document 25.

Section 2.0 of AREVA NP Report No. NGLP/2004/en/0094 provides the description of the tool and its use in the TXS engineering process. Section 2.4.7 provides an example of the methodology for testing a typical function including sample output data. Section 2.5 describes the limitations of testing with SIVAT. Section 4.1 describes the processes and procedures used to develop and test SIVAT under the AREVA NP quality assurance program. Section 4.2 provides a summary of AREVA NP's experience in using SIVAT in the development of TXS application software. The quality assurance process described in Section 4.1 along with the experience documented in Section 4.2 provide basis for AREVA NP's confidence in the use of SIVAT as a V&V tool for TXS application software.

SIVAT, which was developed by AREVA NP, provides a simulation-based test environment for project-related TXS application software being used to validate the application software prior to installation into the target hardware. The validation test cases are created on the basis of the functional requirements defined for the TXS application software. These validation activities serve to validate the detailed software engineering by means of the SPACE graphical editor as well as the application code generation for the project.

Requirements for SIVAT

The basis requirements specified for the development of SIVAT included the following:

- No functional modifications of the original TXS C code for simulation
- Provide proof of correct functioning of the specified TXS I&C system by comparing the simulation results with the Software Requirement Specification
- Support early identification of specification errors in order to reduce effort for correction
- Improve the efficiency of the V&V process within TXS projects.
- Evaluate planned modifications in the application software of installed TXS systems without test field.
- Reproduce events in the simulator that occurred in installed TXS systems.

To attain these goals, SIVAT has the following features:

- Utilization of a modern simulator control system
- Visibility of all signals and variables (up to 400,000)
- Restart capability using Initial Conditions

- Simulation of malfunctions for input/output (I/O) boards, central processor unit (CPU) boards and communication bus failure
- Easy integration of other models (e.g. process models)
- Interface with the dynamic function diagram logic viewer in animated mode to achieve a clear overview of the function diagrams with the actual values during simulation
- No real-time capability (i.e. the simulation can run as fast as possible or can be run in slower time to support visual monitoring)
- Graphical user interface for easy handling of the automatic generation and user friendly interface with the simulation tool
- Script-based simulation to allow reconstruction of simulation cases at any time
- Capability to create individual simulation environments for each project database and user
- Run on a LINUX workstation or a personal computer
- Short time for generation of the simulator models

SIVAT Capabilities

The SIVAT tool utilizes the C Code generated by the SPACE code generators used to generate the code for the target system. The C Code is compiled using two widely used compilers: one for the target processors (Intel iC 86) and one for the simulation environment (GNU). As described in the following extract from AREVA NP Report No. NGLP/2004/en/0094, the code is modified to run as a model in the SIVAT environment, but the code functionality is unaffected.

The TXS validation tool SIVAT uses the SPACE code generators to generate the C codes for the simulation environment. This code is adapted accordingly, such that it can run as a model in the simulation control system. The code functionality remains unaffected. Further libraries (FD, FDG, RTE library) are required in addition to the function block library (FB library) to compile and link the generated C modules (FD, FDG and RTE modules). Thus the following software modules are executing in the simulator:

- The hardware-independent part of the TXS **runtime environment** (includes the runtime environment and the messages sent to other TXS CPUs generated by the RTE code generator).
- The **function diagram group modules**, which call the assigned function diagram modules (FDG modules generated by the FDGM code generator).
- The **function diagram modules**, which in turn call the function blocks from the FB library (FD modules generated by the FDGM code generator).
- The **function blocks** that contain the original, qualified C code of the TXS function blocks that were compiled for the operating system of the simulator computer.

The runtime environment (messages between TXS CPUs and call of the FDG modules), the function diagram group modules, the function diagram modules,

and the function blocks contain the complete I&C functionality of a TXS CPU.
For a TXS CPU model, these components are simulated in SIVAT.

The use of SIVAT was demonstrated during the April 29, 2008, meeting on the Oconee project. The specific changes that are made to the source code for use in the simulator were reviewed during the demonstration. AREVA NP Document 01-5044046-01, "TELEPERM XS SIVAT-TXS Simulation Based Validation Tool (Version 1.5.0 and higher) User Manual (TXS-1047-76-V2.1)," was submitted in an April 3, 2008 letter from Duke Energy to NRC entitled "License Amendment Request for Reactor Protective System/Engineered Safeguards Protective System Digital Upgrade, Technical Specification Change (TSC) Number 2007-09, Supplement 1." It is listed as LAR document 43.

Object code for the target processors is automatically generated from the source code using the qualified code generators in the SPACE tool.

SIVAT is used to support Implementation Activities, as defined in IEEE Std 1012-1998, "IEEE Standard for Software Verification and Validation." The proper focus of the activities is on the source code.

TXS Malfunction Simulation

SIVAT generates three types of functions to simulate TXS malfunctions to verify the effects of certain malfunctions on the specified I&C function:

- Failure of an I/O module
- Failure of a message
- Failure of a complete TXS CPU

These features allow tests to validate that that the response to these faults is as intended.

Limitations of SIVAT Simulation

Section 2.5 of AREVA NP Report No. NGLP/2004/en/0094 describes the limitations of SIVAT simulation. The following system characteristics are not tested by SIVAT:

- Hardware-related parts in the runtime environment (e.g. self-monitoring), interfaces of the runtime environment to the operating system, the operating and network systems themselves, and the I/O module drivers - These components of the TXS system are tested during the TXS generic qualification process. They do not contain any I&C functions that need to be checked system-specifically.
- CPU load - The TXS analysis tool **cpuload** is available for this purpose.
- Network load - The TXS analysis tool **netload** is available for this purpose.

- CPU restart - By setting the malfunction flags for a TXS CPU, the failure of a CPU and its effects on the I&C system can be simulated. This functionality of the TXS system is tested during the TXS generic qualification process.
- Response times - In the real TXS system, all TXS CPUs operate unsynchronized, which results in a system-specific runtime behavior from the acquisition level through processing to the actuation level. In the simulator, all models with the same cycle time are also calculated in the same simulation cycle. The simulation results in an idealized response time behavior. Real response time measurements are validated during system validation testing.

SIVAT Development and Maintenance

SIVAT was developed based on a requirements specification and technical specification document. The development process follows AREVA NP GmbH procedure F-AW-TXS 1.1 "Software Lifecycle Processes." The validation of the product was performed with tests of a real TXS application (data from a test) and the results of a SIVAT simulation of the same application. Changes to the SIVAT tool are controlled via AREVA NP GmbH procedure F-AW-TXS 1.5 "Configuration Management," which establishes requirements to ensure that changes are controlled, documented, and tested.

The TXS configuration management process is described in Engineering Procedure FAW-1.5, "Configuration Management Plan for the TELEPERM XS System Platform." Engineering Procedure FAW-1.5 has evolved since the TXS Topical Report was issued. The changes include the addition of a change control board to the configuration management process and the inclusion of additional detail describing configuration management tasks (e.g., more precise configuration identification). The information security requirements have expanded since the TXS Topical Report was issued. The information security requirements described in Engineering Procedure FAW-1.7, "Information Security," was applicable to TXS activities. These controls were summarized in response to RAI 36.⁷ These development process changes were discussed with NRC during an April 30, 2008, meeting.

NRC also conducted an inspection of the TXS design control process in Erlangen, Germany on March 10-14, 2008. The inspection team reviewed Engineering Procedure FAW-1.5 and its implementation. Particular emphasis was placed on the software configuration change process. No findings were identified against the design control process during the inspection.

SIVAT Validation

Section 4.2 of AREVA NP Report No. NGLP/2004/en/0094 describes the validation efforts for SIVAT.

7 Letter from Ronnie L. Gardner (AREVA NP Inc.) to Document Control Desk (NRC), "Response to Second Request for Additional Information Regarding ANP-10272, 'Software Program Manual for TELEPERM XS™ Safety Systems Topical Report' (TAC No. MD3971)," NRC: 07:029, July 19, 2007.

The I&C system at the Unterweser nuclear power plant was retrofitted with TXS in 1996. At that time, all tests in the test field were still implemented with the real TXS system and a linked process model. The test cases were also verified with a UNISYS test arrangement and a Konvoi system simulator. UNISYS was the simulator control system that was used for TXS simulations prior to SIVAT. The results of this simulation were compared with the test field results and concordance was verified.

No test field was available for upgrading the TXS I&C system at the Unterweser plant in 2000, since the TXS system was installed in the plant. Planned changes could only be tested through validation with SIVAT, which has been available as a TXS V&V tool since 1998. For this purpose, first the test cases from the old simulation environment (UNISYS) and the test field were recalculated with SIVAT. Since the results matched, the verification of the modified I&C functionality was also implemented with SIVAT.

In addition, a closed-loop system test (load shedding from 71% reactor power down to house load) was recalculated by SIVAT and the process model (i.e., system model NLOOP Unterweser). The very high concordance between the actual system behavior and the simulation results lead to the authorization for installing the modified SIVAT-validated TXS I&C. Authorization to install the modified TXS application functions was given based on the very high concordance between the actual system behavior and the SIVAT validation. Plant commissioning took place without findings concerning the new I&C application functions.

A number of test field tests were verified with SIVAT as part of the TXS retrofitting for the Philippsburg 1 nuclear power plant. The very high concordance made it possible to implement individual changes in the TXS configuration even after the test field tests. These modifications were validated and verified exclusively with SIVAT.

Experience with SIVAT

AREVA NP has operating experience with the use of SIVAT for more than 20 project specific applications. No instances have been reported where a system tested using SIVAT did not perform as expected after installation. Section 4.2.3 of AREVA NP Report No. NGLP/2004/en/0094 list nuclear plants projects where SIVAT was used for software validation.

Conformance with IEEE Std 7-4.3.2-2003

IEEE Std 7-4.3.2-2003, "IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations," contains the following guidance for software tools used to support software development processes and V&V processes:

5.3.2 Software tools

Software tools used to support software development processes and verification and validation (V&V) processes shall be controlled under configuration management.

One or both of the following methods shall be used to confirm the software tools are suitable for use:

- a) A test tool validation program shall be developed to provide confidence that the necessary features of the software tool function as required.
- b) The software tool shall be used in a manner such that defects not detected by the software tool will be detected by V&V activities.

Tool operating experience may be used to provide additional confidence in the suitability of a tool, particularly when evaluating the potential for undetected defects.

IEEE Std 7-4.3.2-2003 has been endorsed by NRC in NRC Regulatory Guide 1.152, "Criteria for Use of Computers in Safety Systems of Nuclear Power Plants," Revision 2.

As described above, SIVAT conforms to the guidance in clause 5.3.2 of IEEE Std 7-4.3.2-2003. SIVAT was developed and is maintained using the same TXS development and configuration management process described in the TXS Topical Report. SIVAT was validated against test field data from the TXS retrofit projects at the Unterweser and Philippsburg 1 nuclear power plants. The limitations of SIVAT are clearly identified and understood. The system characteristics not tested by SIVAT are either tested during the TXS generic qualification process, verified with other TXS analysis tools, or validated during system validation testing. AREVA NP has operating experience with the use of SIVAT for more than 20 project specific applications.

Use of SIVAT

In RAI 74, NRC stated that:

The SPM indicates that the SIVAT tool (Simulation and Validation Tool) makes unit and integration tests unnecessary. This approach is unfamiliar to the staff and does not appear to be consistent with industry standards and regulatory guidance.

A similar question was asked during the NRC review of the TXS Software Program Manual. The AREVA NP response was provided in a letter from Ronnie L. Gardner (AREVA NP Inc.) to Document Control Desk (NRC), "Request for Additional Information Regarding ANP-10272, 'Software Program Manual for TELEPERM XS™ Safety Systems Topical Report' (TAC No. MD3971)," NRC: 07:020, on May 22, 2007. The complete response is reproduced below.

AREVA NP Response to RAI 4: In general, the term component is used in ANP-10272 to describe the reusable hardware and software pieces that make up the TXS platform. When the term component is used in the context of IEEE Standard 1012-1998, "Standard for Software Verification and Validation," it would apply from the perspective of TXS Application Software development.

Based on the question, it would have been more appropriate to state in ANP-10272 section 6.0:

One area of exception with regard to the IEEE Standard 1012 is that component verification and validation test execution **of Function Diagrams and Groups of Function Diagram Group Modules** is not considered to be mandatory, but verification of any component testing performed **on the Function Diagrams or Function Diagram Group Modules** is mandatory.

It would be appropriate to consider TXS Function Block testing as the equivalent of component testing. The software life cycle planning process described in section 3 of the TXS topical report applies to the development of the TXS operating system, the function block library for application software, and how it will work on a project-specific basis. It should be noted that TXS software is designed and qualified as an integrated system.

Section 6.2.7.4.1 of ANP-10272 addresses simulation testing, which is an additional layer of development testing performed using SIVAT. This testing is performed on Function Diagrams and Function Diagram Group Modules. SIVAT simulation testing falls between function block testing (a better equivalent to component testing) and the factory acceptance test (FAT), which serves as the integration and system testing.

The terms "software modules" and "subsystems" are generally used to describe groups of software objects. This usage applies a collection of function blocks in the Software Design Document and in the SPACE tool that perform a particular function or set of functions within the application software. This general use of the term could also be interchanged with the terms "Function Diagrams" and "Function Diagram Group Modules" as used in the TXS topical report.

General SIVAT Test Methodology

The general methodology used to develop SIVAT test specifications and test procedures consists of the following eight steps.

1. The project-specific application software functionality that is specified in the Software Requirements Specification is tested to validate that the software elements (e.g., modules, submodules, and functions) correctly implement software requirements. As a minimum the criteria for this determination are:
 - Compliance with functional requirements.
 - Performance at boundaries, interfaces, and under stress and error conditions.
2. The proper functionality of the project-specific application software is tested to validate the following standard TXS characteristics:

- Signals to Output boards must have no fault status at all times, even under error and stress conditions.
 - Test results must be verified from start of test until the completion of the test in order to ensure that no unexpected intermediate results are present.
 - Signals must be handled in a manner to ensure spurious alarms are not generated by the software.
 - Correct setting of function block parameters must be checked against software requirements.
3. The Software Validation Test Plan ensures that the functional requirements of the software design detailed in the Software Requirements Specification are properly implemented in the SPACE application. The comprehensiveness of the testing effort must ensure that all functionality defined in the Software Requirements Specification is tested.
4. Test Specifications are prepared for each Input, Function, and Output Module or Submodule.
- The Test Specifications must ensure that all functionality as defined in the Software Requirements Specification is tested.
 - The Test Specifications must specify test cases that test the combinational logic of the individual software modules and submodules (Input, Function, and Output). Overlapping tests are used to ensure that the boundaries and interfaces of the software modules are fully tested.
 - Test Specifications specify the overall test design and the specific test cases, including detailed instructions for testing and the test case acceptance criteria (i.e., expected results) to be employed during the testing effort.
 - The Test Specifications incorporate the Test-Design Specification and Test-Case Specification into a single document and conform to IEEE Std 829-1983, "IEEE Standard for Software Test Documentation," and IEEE Std 1008-1987, "IEEE Standard for Software Unit Testing."
5. Test Procedures are prepared for each Input, Function, and Output Module or Submodule.
- The Test Procedures contain test scripts that implement the test cases defined in the Test Specifications.
 - The Test Procedure verifies that the correct versions of the project-specific TXS application software, SIVAT, and test scripts are used for the testing.
 - The Test Procedures conform to IEEE Std 829-1983 and IEEE Std 1008-1987.

6. The Test Execution is conducted in a controlled manner.

- Test tasks are executed in the following sequence:
 - Input Submodule Tests - Input Submodules read the input information from the field devices, perform signal validation and failure annunciation functions, and distribute the input signals to the I&C Functions. These tests aim at checking that the correct data is sent to the related I&C Functions.
 - Function Module Tests - I&C Functions obtain data from the Input Submodules, then perform the logic functions and send actuation signals to the Output Submodules or actuation devices. These tests aim at checking that the implemented logic and the correct output to the annunciators, actuators, and to the related Output Submodules.
 - Output Submodule Tests - Output Submodules obtain the outputs from the I&C Functions and then send the actuation signals to the actuation devices. Output Submodules also process checkback signals from the field device and provide component test logics. These tests aim at checking that the implemented logic, the actuation of the field device, and the correct status of the actuated field devices.
- The Signal Monitoring Submodules shall be tested together with its corresponding functions and modules.
- Automated test scripts generate data files capturing the results of the test runs.
- Test data files are entered in the Software Library.
- Test data contained in the data files is plotted using the SIVAT plot conversion tool.
- Data plots are reviewed against the expected results.
- Discrepancies shall be logged for disposition and captured in the Test Incident Report.
- Test Logs and a summary of the executed tests, including discrepancies affecting the software design and implementation, are documented in the Test Summary Report

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

- The Test Item is considered successfully passed when the results of the test match the predicted results described in the Test Specification with no unexpected intermediate results.
- A test Item containing unexpected results may be acceptable considered successfully passed if the evaluation of the unexpected result concludes that the TXS application software is functioning correctly. Disposition/justification of the item is documented and

preserved in the Test Incident Report. Under these conditions, a retest of the item will not be necessary.

- The Test Item is considered failed if the test script has a syntax error that prevents the script from running or if the test script or the Test Specification is found to be in error (i.e., the results of the test do not match the predicted results described in the Test Specification).
- Any errors encountered while performing the test will be documented in the Test Log and Test Incident Report.

8. 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 shall be 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 shall be performed on all sections of the Test Specification determined necessary and recorded in the Test Incident Report.

System Testing

System testing (including FAT) is performed during the Testing Phase. System testing validates that the functionality of the system meets the design and customer requirements in the fully integrated system. Application software validation testing can also be performed as a first step of the system testing (i.e. on the actual equipment) if it was not performed with SIVAT. The additional application software validation testing during system testing validates that the functionality of the application software meets software requirements for its intended use. A FAT is a subset of the system testing that demonstrates to the customer that the finished

system meets the functional and safety requirements. A system test report (including the FAT report) is issued at the end of the testing phase.

Alignment with IEEE Std 1012-1998 Testing Activities

IEEE Std 1012-1998, "IEEE Standard for Software Verification and Validation," 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 Figure 2 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 TXS generic qualification testing and project-specific testing addresses all of the testing activities in IEEE Std 1012-1998, as shown in the following table.

Table 73-1 Alignment with IEEE Std 1012-1998 Testing Activities

IEEE Std 1012-1998 Testing Activity	Generic TXS Testing	Project-Specific Testing
Component Testing	X (hardware and software type tests, including Function Blocks)	Not Applicable (based on use of qualified hardware and software modules)
Integration Testing	X	Application Software: SIVAT for integration of Function Block modules Optional X (see Note 1)
		System Components: Pre-FAT prerequisites and procedure dry runs (manufacturing tests)
System Testing	X	X (integrated in system testing, including FAT, based on use of qualified system components and development tools)
Acceptance Testing	Not Applicable	

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

Note 1 – Additional application software integration and functional test cases are added to the scope of system validation testing for the case where SIVAT testing is not used for application software integration and functional testing to satisfy IEEE Std 1012-1998 validation requirements.

Conclusions

SIVAT provides a simulation-based test environment for project-related TXS application software being used to validate the application software prior to installation into the target hardware. The validation test cases are created on the basis of the Software Requirements Specification defined for the TXS application software. These testing activities serve to validate the detailed software engineering by means of the SPACE graphical editor as well as the application code generation for the project.

SIVAT conforms to the guidance in clause 5.3.2 of IEEE Std 7-4.3.2-2003.

The combination of TXS generic qualification testing and project-specific testing addresses all of the testing activities in IEEE Std 1012-1998.

Software Program Manual Changes

The Software Program Manual will be revised to incorporate this information.

RAI-74 Question: How does the use of SIVAT:

Ensure a high-quality test process and safety of the resultant system which is equivalent or better than traditional unit and integration testing

and

Demonstrate that the system requirements specifications have been correctly translated into error-free application code?

NRC Concern: The SPM indicates that the SIVAT tool (Simulation and Validation Tool) makes unit and integration tests unnecessary. This approach is unfamiliar to the staff and does not appear to be consistent with industry standards and regulatory guidance.

The use of the SIVAT tool was not identified in the TXS topical report, and it is not clear if the software tested by SIVAT is the same compiled application code to be loaded, unaltered, onto TXS hardware. It appears that the first time the compiled operational code is tested is during the FAT, which is developed by the design and test group, not the V&V Group.

The staff does not understand, based on the limited information submitted describing SIVAT and the V&V process, how software testing using SIVAT can demonstrate that the system requirements specifications have been correctly translated into error-free application code. The staff believes that testing performed by unit and integration tests should be performed on the actual operational code, and therefore it may be necessary to perform additional software testing.

In addition, it should be understood that test plans and procedures generation and verification are the sole responsibility of the V&V team. The existence of an automated tool does not relieve the V&V team of their responsibilities.

AREVA NP's response should support a conclusion that the SIVAT testing will provide confidence in a high quality test process and equivalent confidence in the safety of the resultant system.

Demonstration of the SIVAT tool and associated development and V&V tools may contribute to the NRC staff's confidence and understanding of AREVA NP's approach as outlined in the SPM. Arrangement of such a demonstration may be coordinated through the NRC's EPR Projects Branch.

Applicant Reference:

ANP-10272

Software Safety Plan (Page 4-1)

... AREVA NP uses SIVAT testing of the application software generated by the SPACE tool to detect errors that would prevent the software from fulfilling its safety function. SIVAT testing,

coupled with the FMEA, response time analysis, and FAT are sufficient to ensure that there are no software hazards...

AREVA NP Response to RAI 74: See response to RAI 73.

RAI-75 Question: Has the SIVAT tool been qualified in a manner similar to that required for software performing safety-related functions, and does the software lifecycle process of the SIVAT tool development meet the requirements for that type of software?

NRC Concern: The SPM describes the SIVAT tool as a key component in the application software V&V testing. Since the safety and quality of the resultant application software is paramount, tools used to assure that quality and safety should be, in of themselves, high quality. No such demonstration of quality has been made for the SIVAT tool.

The TELEPERM XS and SPM topical reports lack details about the qualification of the SIVAT software and the process quality used in the SIVAT tool development.

The TELEPERM XS and SPM topical reports do not identify how SIVAT and V&V procedures will be used in a combined manner such that defects not detected by the SIVAT tool will be detected by V&V activities.

AREVA is requested to submit additional information to support any qualification claims for the SIVAT tool.

Applicant Reference:

ANP-10272

AREVA NP Response to RAI 75: See response to RAI 73.

RAI-76 Question: According to the SPM, is the QA Manager responsible for determining if the QA procedures are adequate?

If not, to what extent is that responsibility transferred, and what is the justification for giving the Technical Manager this responsibility?

NRC Concern: The language of the SPM seems to indicate that the Technical Manager is assuming the responsibilities of the Quality Assurance (QA) Manager. If this is the case, then AREVA NP would have to indicate the extent to which the QA Manager responsibilities have been transferred to the Technical Manager and justify how this maintains a high-quality process. The NRC would then determine the acceptability of such a justification.

However, if the intent of the language in the SPM is to indicate that the Technical Manager is responsible for ensuring that his group follows the QA plans, while the QA Manager retains responsibility for managing and ensuring the adequacy of processes and procedures in those plans, the language of the SPM should be refined to more clearly indicate this standard approach. Submit any page changes needed for this clarification.

Applicant Reference:

ANP-10272

Technical Manager (Page 2-2)

Under the implementing procedures of the AREVA NP Quality Management Manual, the technical manager is responsible for ensuring that the applicable QA processes and procedures are implemented on all projects.

Management (Page 3-1)

The technical manager manages the Software Quality Assurance Plan. The QA group verifies that the implementation of QA requirements is in accordance with the Quality Management Manual. The technical manager ensures that software and associated documentation has been developed in accordance with the Software Quality Assurance Plan, which includes ensuring that the testing and documentation requirements established in the test plan have been followed.

AREVA NP Response to RAI 76: The sentence from Section 2.1.1 will be revised as follows:

The technical manager is responsible for performing all project work in accordance with the applicable QA processes and procedures, as required by the AREVA NP Quality Management Manual.

Clause 4.3.1 of IEEE Std 730-2002, "Standard for Software Quality Assurance Plans," states:

4.3.1 Organization. This section shall depict the organizational structure that influences and controls the quality of the software. This shall include a description of each major element of the organization together with the roles and delegated responsibilities. The amount of organizational freedom and objectivity to evaluate and monitor the quality of the software, and to verify problem resolutions, shall be clearly described and documented. In addition, the organization responsible for preparing and maintaining the SQAP shall be identified.

NRC has not endorsed IEEE 730; therefore, no additional guidance is available regarding the SQAP or the interpretation of IEEE 730.

The information from Section 3.2 referenced in the RAI is a statement that addresses the requirement in the last sentence of clause 4.3.1. The AREVA NP Quality Assurance organization approves the SQAP.

RAI-77 Question: According to the SPM, is the V&V Manager responsible for the disposition of discrepancy reports by ensuring that the actions taken and changes made in such disposition are correct, appropriate, and sufficient?

If not, what is the justification for giving the Technical Manager this responsibility?

NRC Concern: The language of the SPM seems to indicate that the Technical Manager is assuming the responsibilities of the V&V Manager. If this is the case, then AREVA NP would have to indicate the extent to which the V&V Manager responsibilities have been transferred to the Technical Manager and justify how this maintains a high-quality, verified and validated process. The NRC would then determine the acceptability of such a justification, though it is likely that such a deviation from standard V&V practices would be found unacceptable.

If the SPM indicates the V&V Manager is, in fact, responsible for such disposition, the SPM language should be refined to clearly indicate this standard approach. Submit any page changes needed for this clarification.

Applicant Reference:

ANP-10272

Technical Manager (Page 2-2)

The technical manager is responsible for disposition of discrepancy reports and other anomalies generated in the course of verification and validation.

AREVA NP Response to RAI 77: The sentence from Section 2.1.1 will be revised as follows:

The Technical Manager is responsible for correcting design errors and making software changes associated with discrepancies and other anomalies identified by verification and validation activities.

The V&V organization is responsible for correcting errors in validation test documents, since they are responsible for those documents. The V&V organization reports on anomalies identified as well as the resolution of the anomalies, as specified in IEEE Std 1012-1998. No exceptions are taken to this reporting responsibility.

RAI-78 Question: What is AREVA NP's justification in reducing the scope of the software integration effort? In answering this question, please also address:

What is the specific reduction in scope of integration efforts?

What functionality of the SPACE tool, or effort occurring during the SPACE tool development, is considered to alleviate the need for separate software integration testing?

NRC Concern: AREVA NP has not provided enough information explaining what is considered to be the reduced scope of integration testing.

AREVA NP has not provided justification for reducing the scope or entirely eliminating software integration testing.

AREVA NP has not proposed a sufficient explanation as to why the SPACE tool surpasses integration testing. Neither has the functionality of the SPACE tool been described to an extent that builds the case for its use instead of integration testing.

Submit the appropriate page changes in the SPM to support the justification.

Applicant Reference:

ANP-10272

I&C Engineers (Page 2-4)

Because the application software is generated by the SPACE tool and the SPACE tool is designed to provide the software to run on the TELEPERM XS system software, no separate integration effort for this software is required.

AREVA NP Response to RAI 78: See response to RAI 73.

RAI-79 Question: Does the V&V team

Ensure that the outputs of each phase of the design process fulfill the requirements of each previous phase,

and

Determine that the design outputs comply with functional, performance, and interface requirements through tests and inspections?

NRC Concern: The language of the SPM seems to indicate that the responsibility of the V&V team is limited to the traceability analysis and the functional requirements specification (FRS) review. The SPM does not indicate that the V&V team will fulfill the V&V team responsibilities as described by relevant IEEE standards.

IEEE 1012 defines the following:

verification:

(A) *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. (B) The process of providing objective evidence that the software and its associated products conform to requirements (e.g., for correctness, completeness, consistency, accuracy) for all life cycle activities during each life cycle process (acquisition, supply, development, operation, and maintenance); satisfy standards, practices, and conventions during life cycle processes; and successfully complete each life cycle activity and satisfy all the criteria for initiating succeeding life cycle activities (e.g., building the software correctly).*

validation:

(A) *The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. (B) The process of providing evidence that the software and its associated products satisfy system requirements allocated to software at the end of each life cycle activity, solve the right problem (e.g., correctly model physical laws, implement business rules, use the proper system assumptions), and satisfy intended use and user needs.*

IEEE 100, The Authoritative Dictionary of IEEE Standards Terms, provides more concise definitions:

verification:

The process of determining whether or not the product of each phase of the digital computer system development process fulfills all the requirements imposed by the previous phase.

validation:

The test and evaluation of the integrated computer system to ensure compliance with the functional, performance, and interface requirements.

verification and validation:

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.

Any deviation from these definitions in the responsibilities or methods of the V&V team should be identified and justification should be provided for those deviations, including necessary page changes to the SPM.

Applicant Reference:

ANP-10272

Verification and Validation Team (Page 2-5)

The verification and validation team performs verification reviews of the FRS and the traceability analysis of the SRS into design and test plans.

AREVA NP Response to RAI 79: The Verification and Validation group performs the Activities and Tasks listed in Table 1 of IEEE Std 1012-1998, IEEE Std 1012-1998, "Standard for Software Verification and Validation," as modified by 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," for the scope of TXS project activities governed by the Software Program Manual.

The verification and validation activities defined in the Software Program Manual for a TXS project end with the FAT. Subsequent installation activities, starting with the site acceptance test, are the primary responsibility of the customer. AREVA NP support for installation activities are defined by the customer for each project.

The process for post-FAT software changes is described in Section 7 of the Software Program Manual.

As a point of clarification, the actual definitions of verification and validation endorsed by Regulatory Guide 1.168, Revision 1, are:

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

3. Definitions, abbreviations, and conventions

3.1 Definitions

The following terms, including those defined in other standards, are used **as** indicated in this standard. Annex I contains definitions taken from other existing standards.

....

3.1.31 validation: See Annex I,

3.1.32 verification: See Annex I.

Annex I (normative) Definitions from existing standards normative)

The following are definitions from existing standards as identified in the brackets []. These definitions are placed in this annex so that the body of this standard will not require updating in the event the cited standards and their definitions change.

...

validation: Confirmation by examination and provisions of objective evidence that the particular requirements for a specific intended use are fulfilled.

NOTES

- 1-In design and development, validation concerns the process of examining a product to determine conformity with user needs.
- 2-Validation is normally performed on the final product under defined operating conditions. It may be necessary in earlier stages.
- 3-"Validated" is used to designate the corresponding status.
- 4- Multiple validations may be carried out if there are different intended uses. [ISO 8402:1994]

...

verification: Confirmation by examination and provisions of objective evidence that specified requirements have been fulfilled.

NOTES

- 1-In design and development, verification concerns the process of examining the result of a given activity to determine conformity with the stated requirement for that activity.
- 2-"Verified" is used to designate the corresponding status. [ISO 8402:1994]

AREVA NP agrees with NRC that the definitions in IEEE Std 1012 are not useful. The Software Program Manual uses definitions from IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology

validation. 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. 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.*

verification and validation (V&V). 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.

RAI-80 Question: Does AREVA NP intend that where the SPM references other material in support of fulfilling requirements, that material would not be subject to review and inspection by the NRC in making any determination of conformance, safety, and acceptability for any given application, design certification, or amendment?

NRC Concern: The SPM sets forth requirements and objectives for various software lifecycle and other plans, and in many places it references other documents to make claims of full conformance to guidance, regulations, and standards. While the sum of these pieces may ultimately be found acceptable for a specific application, it cannot be assumed that the approval of one high-level document (SPM) cascades to its subordinate and referenced documents. Every document referenced in support of the SPM and the plans it describes must be considered on its own merit.

One such example is the reference of the AREVA NP Quality Management Manual. AREVA NP indicates that the Software Quality Assurance Plan does not fulfill IEEE 730 requirements on its own, but in combination with the AREVA NP Quality Management Manual and QA reviews and audits. AREVA NP does not delineate which portions of IEEE 730 are covered by the SQAP and which portions are covered by the Quality Management Manual, nor does AREVA NP provide the Quality Management Manual for review.

For any license amendment request, combined license application, or design certification referencing the SPM in support of the SQAP, the NRC may have to review, inspect, or audit; and find acceptable; the Quality Management Manual and associated operating instructions. This example extends to all of the plans described by SPM.

The NRC does not request submission of the five referenced plans or the AREVA NP Quality Management Manual at this time.

Applicant Reference:

ANP-10272

Introduction (Page 1-1, 2)

In addition to this Software Program Manual, the program consists of the following plans:

- 1. 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.*
- 2. 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.*
- 3. Software Verification and Validation Plan, which describes the method that ensures correctness of the software.*

4. *Software Configuration Management Plan, which describes the method that maintains the software in a controlled configuration at all times.*

5. *Software Operations and Maintenance Plan, which describes post-customer delivery software practices.*

The combination of the Software Program Manual and the five plans above constitute a program that conforms to the guidance of Nuclear Regulatory Commission (NRC) Branch Technical Position (BTP) Human, Instrumentation and Controls Branch Topical (HICB)-14 (Reference 11)...

*...The Software Program Manual establishes the requirements and objectives for the Software Quality Assurance Plan, Software Safety Plan, Software Verification and Validation Plan, Software Configuration Management Plan, and Software Operations and Maintenance Plan. These five plans are implemented as AREVA NP operating instructions and will conform to the requirements established in the Software Program Manual. **In some cases additional operating instructions will be 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 five plans are available onsite at AREVA NP facilities to support NRC review of this topical report.*

Purpose (Page 3-1)

*The Software Quality Assurance Plan fulfills the requirements for a software quality assurance plan in accordance with IEEE 730 (Reference 17) **but must be considered along with the AREVA NP Quality Management Manual and the Quality Assurance reviews and audits for complete fulfillment of the IEEE requirements.***

AREVA NP Response to RAI 80: AREVA NP understands that NRC may choose to review, audit, or inspect information associated with any future licensing application involving TXS technology.

AREVA NP also acknowledges and appreciates the NRC review and comments on the specific documents involved with the December 19, 2007, audit. The disposition of the comments provided to AREVA NP was documented in a letter from Ronnie L. Gardner (AREVA NP Inc.) to Document Control Desk (NRC), "Follow-up Actions from December 19, 2007, NRC Audit of ANP-10272, "Software Program Manual for TELEPERM XS Safety Systems Topical Report," Program Implementation (TAC No. MD3971)" NRC: 08:008, January 24, 2008.

Information regarding the AREVA NP quality assurance program for was previously provided in response to RAI 31.⁸ AREVA NP's implementation of the QMM is periodically audited by the Nuclear Procurement Issues Committee (NUPIC). The NUPIC program evaluates suppliers

8 Letter from Ronnie L. Gardner (AREVA NP Inc.) to Document Control Desk (NRC), "Response to Second Request for Additional Information Regarding ANP-10272, 'Software Program Manual for TELEPERM XS™ Safety Systems Topical Report' (TAC No. MD3971)," NRC: 07:029, July 19, 2007.

furnishing safety-related components and services and commercial grade items to nuclear utilities. The most recent NUPIC audit of AREVA NP was performed in November 2006.

The QMM also allows for the issuance of quality assurance plans to augment the quality requirements for a specific customer or project. AREVA NP Topical report ANP-10266A, Revision 01, "AREVA NP Inc. Quality Assurance Plan (QAP) for Design Certification of the U.S. EPR," (referred to as the U.S. EPR Quality Assurance Plan) was issued to describe the Quality Assurance Plan applicable to the Design Certification of the U.S. EPR. The plan is based on the eighteen point criteria of 10 CFR 50, Appendix B, and ANSI/ASME NQA-1-1994. NRC issued a safety evaluation report for this topical report in a letter from Getachew Tesfaye to Ronnie L. Gardner dated April 26, 2007.

RAI-81 Question: What is the extent of the project manager's assessment of software safety risk in accordance with IEEE 7-4.3.2?

NRC Concern: The language in the SPM indicates that the project manager assesses technical, schedule, and regulatory risks of software projects. It is unclear what the SPM means by "technical" risks, as it is not defined nor is it a common term used in IEEE 7-4.3.2. Schedule and regulatory risks, while of concern to an operating business, are not of interest to the NRC in making safety determinations. This section of the SPM does not emphasize the safety risk of the project.

A project manager's assessment of software safety risk should take into account product engineering, development environment and program constraints, system risks, mechanical/electrical hardware integration, risks due to size and complexity of the product, the use of pre-developed software, risks from program interfaces, maintenance risk, security risk, and the risk associated with each V&V task; and should follow the requirements of IEEE 7-4.3.2.

IEEE 7-4.3.2 contains a section on software project risk management (5.3.6) which defines a concept and outlines the appropriate steps to take in analyzing and implementing risk management. If this is what AREVA NP intends by use of the term "technical" risk, the language of the SPM should be clarified to represent this intent.

If AREVA NP intends a different interpretation, a justification for deviating from IEEE 7-4.3.2 should be made. Submit any page changes necessary for that justification.

An excerpt from IEEE 7-4.3.2, Section 5.3.6 follows:

Software project risk management is a tool for problem prevention: identifying potential problems, assessing their impact, and determining which potential problems must be addressed to assure that software quality goals are achieved. Risk management shall be performed at all levels of the digital system project to provide adequate coverage for each potential problem area. Software project risks may include technical, schedule, or resource-related risks that could compromise software quality goals, and thereby affect the ability of the safety computer system to perform safety related functions. Software project risk management differs from hazard analysis, as defined in 3.1.31, in that hazard analysis is focused solely on the technical aspects of system failure mechanisms.

Risk management shall include the following steps:

- a) Determine the scope of risk management to be performed for the digital system.*
- a. Define and implement appropriate risk management strategies.*
- b. Identify risks to the software project in the project risk management strategy and as they develop during the conduct of the project.*
- c. Analyze risks to determine the priority for their mitigation.*
- d. Develop risk mitigation plans for risks that have the potential to significantly impact software quality goals, with appropriate metrics for tracking resolution progress. (These risks may include technical, schedule, or resource-related*

project risks that could compromise the ability of the safety computer system to perform safety related functions.)

- e. Take corrective actions when expected quality is not achieved.*
- f. Establish a project environment that supports effective communications between individuals and groups for the resolution of software project risks.*

Applicant Reference:

ANP-10272

Risk Management (Page 3-10)

The project manager identifies and assesses the technical, schedule, and regulatory risks of the project.

AREVA NP Response to RAI 81: The Project Manager uses AREVA NP's standardized project management risk assessment tools to assess project risks. Project Management Guideline PMG-7 documents the method and procedure to identify, assess, monitor, and control areas of risk that arise during the software development project. This methodology is used to identify, assess, monitor, and control areas of risk that arise during the software development project. The risk management process is comprised of the following steps; risk identification, analysis and prioritization, response development, and risk monitoring and control. The methodology utilizes a process to rate the complexity and risks of projects to optimize project planning and execution. In the course of project execution, the project risks are monitored, and the original rating is reviewed to determine if the rating needs to be modified.

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

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

The AREVA NP risk management process conforms to the guidance in IEEE 7-4.3.2 clause 5.3.6.

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.

Standard TXS Risk Mitigation Measures

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

The TXS technology is a mature and fully integrated nuclear safety system. The TXS hardware is fully qualified safety-related equipment. The TXS operating system software and Function Block library are developed and maintained using the process described in the TXS Topical Report. The application software is generated by the SPACE tool. The generic TXS qualification process removes risks associated with integration of TXS software and hardware. Design features of the TXS system that address communication independence are addressed in Sections 2.4 and 2.9 of the TXS Topical Report.

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

This Software Program Manual describes the program measures incorporated at AREVA NP to ensure that the TXS 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 that could defeat hardware redundancy. A Defense-in-Depth and Diversity Analysis is performed to ensure that adequate defense-in-depth has been provided in the design. The Defense-in-Depth and Diversity Analysis addresses residual software risks by addressing mitigation of assumed software common mode failures.

Software Program Manual Changes

The Software Program Manual will be revised to incorporate this information.

RAI-82 Question: Does the SPM indicate full compliance with Position 3 of the SRM to SECY 93-087 (software common-mode failure), and which AREVA NP document details that compliance?

NRC Concern: The SPM states that "...the FMEA does not need to consider the effects of a software common-mode failure because this kind of failure is handled by the diversity and defense-in-depth analysis..." This statement is incorrect. Position 3 of the SRM to SECY 93-087 specifically requires that the licensee consider a postulated common-mode failure. Furthermore, the purpose of the D3 analysis, in part, is to support the findings in the FMEA.

If the intent of the SPM is, however, that software common mode failure is considered elsewhere, the language of the SPM should be revised to clearly reflect where it is considered. Submit any page changes necessary to reflect this clarification.

One example of such a clarification could be:

"However, the FMEA does not need to consider the effects of a software common mode failure because this kind of failure is considered in ... to assure that the plant specific diversity and defense-in-depth will handle the postulated software common mode failure."

Applicant Reference:

ANP-10272

Failure Modes and Effects Analysis (Page 4-3)

However, the FMEA does not need to consider the effects of a software common mode failure because this kind of failure is handled by the diversity and defense-in-depth analysis discussed in Section 4.3.1.

AREVA NP Response to RAI 82: The Failure Modes and Effects Analysis (FMEA) is performed to examine 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, "Application of the Single-Failure Criterion to Nuclear Power Generating Station Safety Systems," as endorsed by Regulatory Guide (RG) 1.53, Revision 2, "Application of the Single-Failure Criterion to Safety Systems." AREVA NP meets the requirements of IEEE Std 379-2000 to establish conformance with the requirements of IEEE Std 603-1991, "Criteria for Safety Systems in Nuclear Power Plants," specifically the single-failure criterion as stated in clause 5.1. This use of IEEE Std 379-2000 is consistent with the Regulatory Position of RG 1.53, which states that:

Conformance with the requirements of IEEE Std 379-2000, "Application of the Single-Failure Criterion to Nuclear Power Generating Station Safety Systems," provides methods acceptable to the NRC staff for satisfying the NRC's regulations with respect to the application of the single-failure criterion to the

electrical power, instrumentation, and control portions of nuclear power plant safety systems.

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 379.

IEEE 379 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 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 FEMA analysis. Extended FMEA analyses are not requirements of IEEE 379 and are not required to establish conformance with the requirements of IEEE 603. Instead, extended FMEA analyses, when performed, are used to provide additional insights regarding risk, reliability, or other performance objectives specified by the customer.

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 effects of a software common mode failure are addressed by the diversity and defense-in-depth analysis required by Branch Technical Position 7-19, "Guidance for Evaluation of Diversity and Defense-in-Depth in Digital Computer-Based Instrumentation and Control Systems," as noted in Section 4.3.2 of the Software Program Manual. This position is consistent with Item 7 of DI&C-ISG-02, "Task Working Group #2: Diversity and Defense-in-Depth Issues - Interim Staff Guidance," Revision 1.

Software Program Manual Changes

The Software Program Manual will be revised to incorporate this clarification.

RAI-83 Question: Does the SPM indicate by stating that a software safety organization is not necessary that a software hazard analysis is also not necessary? If so, what justification, beyond what is already provided in the SPM, does AREVA NP propose for the elimination of the software hazard analysis? If AREVA NP does intend to perform a software hazard analysis, what language indicates this in the SPM?

NRC Concern: The SPM language seems to indicate that test, FMEA, response time analysis, and FAT are sufficient to eliminate the need for a software hazards analysis. If this were the case, any system with adequate test, analysis, and V&V would be exempt from the need for the software hazards analysis. This, however, is not the case.

The software hazards analysis is used to look at failures such as software malfunctions which could defeat the safety function, failures induced by use of out-dated procedures, system I/O incompatibilities (electrical/mechanical) with plant interfaces, and failures with hard to notice or no indications, and so on.

Often, a software hazards analysis is not a separate analysis, but part of a larger safety analysis or some other analysis which specifically looks for software hazards. If AREVA NP has included the software hazards analysis as part of some other analysis, the SPM language should be revised to clarify that inclusion and state where the software hazard analysis occurs.

If, however, AREVA NP intends to deviate from having a standard software hazard analysis due to the use of SIVAT as a V&V tool, the quality and capability of the SIVAT tool would have to be demonstrated and the NRC would have to find it of a quality suitable for use in safety-related applications to consider this alternative.

Applicant Reference:

ANP-10272

Software Test Report on SIVAT Testing (Page 4-4)

The SIVAT tool tests the functionality of the software and provides the results. The verification and validation organization reviews the results of the simulation testing. This approach is different than the guidance of BTP HICB-14. AREVA NP concluded that an independent software safety organization is not necessary to perform this testing. Independent reviews of the work done with SPACE and SIVAT performed by the verification and validation organization, coupled with the FMEA, response time analysis, and FAT are sufficient to ensure that there are no software hazards.

AREVA NP Response to RAI 83: The Technical Manager is responsible for the execution of the Software Safety Plan. Various parts of the project organization perform the specific software safety analyses. 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.

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, "IEEE Standard for Software Safety Plans." 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 TXS application software, the code is automatically generated by the SPACE tool. As such, the Software Design group does not create code; instead, it is involved in many of the software safety analyses. . The responsibility to produce a safe TXS 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.

Software Safety Analyses

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

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

The software hazards associated with first-of-a-kind engineering work are minimized through the use of qualified software development tools and structured engineering analyses. The use of a qualified Function Block library provides a large experience base for the standard modules. The use of the object-oriented automated code generation tool (SPACE) eliminates an important human error source by eliminating conventional software development and code generation. The SPACE tool eliminates both errors of translation and the introduction of complexity by engineers trying to optimize application coding. The use of the SPACE tool supports the development of high quality software with a less complex process, which eliminates many software hazards associated with manual coding.

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

The generic TXS foundation is supplemented with project-specific analyses that address the first-of-a-kind engineering for a TXS project. The following sections describe the project-specific

safety analyses that are performed to ensure that the application software satisfies the design basis safety requirements. These activities ensure the high reliability necessary for safety-related software in a safety-related system. When combined together these activities also satisfy the requirements for a software hazards analysis and meet the intent of IEEE 1228.

- Preliminary Hazard Analysis - The generic safety assessment of the TXS platform described in the TXS Topical Report and the project-specific development process described in the Software Program Manual coupled with the performance of a project-specific diversity and defense-in-depth analysis satisfy the requirement for a preliminary hazards analysis.
- Diversity and Defense-in-Depth Analysis - The diversity and defense-in-depth addresses residual software hazards by addressing mitigation of assumed software common mode failures.
- 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 and support independent verification and validation activities.
- Failure Modes and Effects Analysis – This analysis ensures that the single failure requirements associated with system safety analysis requirements and assumptions are satisfied.
- Response Time Analysis - The response time analysis ensures that the system safety analysis requirements and assumptions are satisfied.
- Verification and Validation Activities - 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 Integration Testing - The testing validates that the right software modules have been properly used and that the functionality of the application software meets the software requirements and customer specifications. SIVAT can simulate various TXS malfunctions to verify that the response to these faults is as intended.
- Criticality Analysis - Verification and validation activities based on the Software Integrity Level assignment applied to non-safety software eliminate software hazards associated with the non-safety to safety interconnections.
- Factory Acceptance Testing - The FAT validates that the functionality of the system meets the system requirements in the fully integrated system. Application software integration and functional testing can also be performed during FAT if it was not performed with SIVAT. The additional application software integration and functional

testing during FAT validates that the functionality of the application software meets software requirements.

Software Program Manual Changes

The Software Program Manual will be revised to incorporate this information.

RAI-84 Question: What is AREVA NP's justification for not using a software configuration management organization or software configuration control board for software configuration management?

NRC Concern: There is no justification given for not having a configuration control board, nor is a set of software configuration management controls and procedures provided or described. A software configuration management board is specifically relied upon to ensure that all configuration changes are adequately and appropriately justified, tested, and documented, and that the urgencies of the project cost and schedule do not permit changes to be made without this justification, test, and documentation.

Applicant Reference:

Organization (Page 5-2)

The software engineering group performs the software configuration management activities described in this Software Configuration Management Plan. The software supervisor is responsible for these activities. As such, no separate software configuration management organization is required for the implementation of the software configuration management activities on a TELEPERM XS software project. The organization is as described in Section 2.0 above."

Configuration Control Boards (Page 5-4)

AREVA NP does not use configuration control boards for software configuration management.

AREVA NP Response to RAI 84: AREVA NP does not use Configuration Control Boards for the development of TXS application software; however, the overall AREVA NP approach to configuration management of the TXS platform, TXS projects, and the project-specific application software meets the intent of IEEE Std 828-1990 and IEEE Std 1042-1987.

NRC Regulatory Guide 1.169, September 1997, "Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants," endorses IEEE Std 828-1990, "Standard for Software Configuration Management Plans," and IEEE Std 1042-1987, "Guide to Software Configuration Management." The Regulatory Guide does not mention configuration control boards or elaborate or interpret the guidance in the endorsed IEEE standards.

AREVA NP's interpretation of IEEE Std 828-1990 is that it specifies requirements for a configuration control board, if one is used. Specifically, clause 2.2.2 states that

For any review board or special organization established for performing SCM activities on this project, the Plan shall describe its ... (responsibilities listed).

Similarly, clause 2.3.2.3 states that:

The Plan shall identify each configuration control board (CCB) and its level of authority for approving proposed changes. A CCB may be an individual or a group. Multiple levels of CCBs may be specified, depending upon the degree of system or project complexity and upon the project baseline involved. When multiple CCBs are used, the Plan shall specify how the proper level is determined for a change request, including any variations during the project life cycle.

For any CCB utilized, the Plan shall indicate its level of authority and its responsibilities as defined in 2.2.2.

And, finally, clause 2.3.5 states that:

For any CCB established to control interfaces, the Plan shall identify its responsibilities and procedures as specified in 2.2.2.

IEEE Std 1042-1987 provides insight as to the underlying purpose of configuration control boards in clause 2.3.3.

Another functional concept of SCM is the extended use of configuration control boards (CCB). This concept provides for implementing change controls at optimum levels of authority. Configuration control boards can exist in a hierarchical fashion (for example, at the program, system design, and program product level, or one such board may be constituted with authority over all levels of the change process. In most projects, the CCB is composed of senior level managers. They include representatives from the major software, hardware, test, engineering, and support organizations. The purpose of the CCB is to control major issues such as schedule, function, and configuration of the system as a whole.

The more technical issues that do not relate to performance, cost, schedule, etc, are often assigned to a software configuration control board (SCCB). The SCCB discusses issues related to specific schedules for partial functions, interim delivery dates, common data structures, design changes, and the like. This is the place for decision-making concerning the items that must be coordinated across CI but which do not require the attention of high level management. The SCCB members should be technically well-versed in the details of their area; the CCB members are more concerned with broad management issues facing the project as a whole and with customer issues.

AREVA NP does not use Configuration Control Boards for the development of TXS application software.

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

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

The intent of the high-level project-related review is addressed the by the routine project management meetings established in the Project Plan. These project meetings include internal project meetings, customer interface meetings, and management oversight meetings and involve the project stakeholders.

The intent of the low-level Software Configuration Control Board is met by the Design Review Board (briefly described in Software Program Manual sections 3.4.2 and 10.3) which is used to review TXS project design issues.

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

A separate Configuration Control Board to address changes to the application software for a TXS project would be redundant to the project meetings and Design Review Boards. Members of a separate Configuration Control Board would include the project team members that participate in the other forums and interact with each other on a daily basis. Since all changes are tracked via the Open Item process, and that process requires an evaluation of document and software changes, such a separate Configuration Control Board would be a duplication of other existing processes using the same personnel.

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

The overall AREVA NP approach to configuration management of the TXS platform, TXS projects, and the project-specific application software meets the intent of IEEE Std 828-1990 and IEEE Std 1042-1987.

Software Program Manual Changes

The Software Program Manual will be revised to incorporate this information.

RAI-85 Question: What is AREVA NP's justification for taking exception to IEEE 1012 in reducing the scope of component V&V? How does AREVA NP determine what will undergo component V&V and what will not?

NRC Concern: AREVA NP proposes an exception to IEEE 1012 that has not been adequately justified. Furthermore, the differentiation of what is to undergo component V&V and what is not has not been described—no methodology for making that determination has been proposed.

The NRC cannot make a determination as to the acceptability of this deviation or exception based on the limited information provided by AREVA NP. AREVA NP will have to provide additional information to assure that the deviation from IEEE 1012 will support an equivalent level of resultant software safety or indicate full conformance with IEEE 1012.

Applicant Reference:

Software Verification and Validation Plan (Page 6-1)

*The Software Verification and Validation Plan follows the guidance of the applicable recommendations of IEEE 1012, which is endorsed by Regulatory Guide 1.168. **One area of exception with regard to the IEEE Standard 1012 is that component verification and validation test execution is not considered to be mandatory, but verification of any component testing performed is mandatory.***

AREVA NP Response to RAI 85: A similar question was previously asked during the NRC review of the TXS Software Program Manual. The AREVA NP response was provided in a Letter from Ronnie L. Gardner (AREVA NP Inc.) to Document Control Desk (NRC), "Request for Additional Information Regarding ANP-10272, 'Software Program Manual for TELEPERM XS™ Safety Systems Topical Report' (TAC No. MD3971)," NRC: 07:020, on May 22, 2007. The complete response is reproduced below.

AREVA NP Response to RAI 4: In general, the term component is used in ANP-10272 to describe the reusable hardware and software pieces that make up the TXS platform. When the term component is used in the context of IEEE Standard 1012-1998, "Standard for Software Verification and Validation," it would apply from the perspective of TXS Application Software development.

Based on the question, it would have been more appropriate to state in ANP-10272 section 6.0:

One area of exception with regard to the IEEE Standard 1012 is that component verification and validation test execution **of Function Diagrams and Groups of Function Diagram Group Modules** is not considered to be mandatory, but verification of any component testing performed **on the Function Diagrams or Function Diagram Group Modules** is mandatory.

It would be appropriate to consider TXS Function Block testing as the equivalent of component testing. The software life cycle planning process described in section 3 of the TXS topical report applies to the development of the TXS operating system, the function block library for application software, and how it will work on a project-specific basis. It should be noted that TXS software is designed and qualified as an integrated system.

Section 6.2.7.4.1 of ANP-10272 addresses simulation testing, which is an additional layer of development testing performed using SIVAT. This testing is performed on Function Diagrams and Function Diagram Group Modules. SIVAT simulation testing falls between function block testing (a better equivalent to component testing) and the factory acceptance test (FAT), which serves as the integration and system testing.

The terms "software modules" and "subsystems" are generally used to describe groups of software objects. This usage applies a collection of function blocks in the Software Design Document and in the SPACE tool that perform a particular function or set of functions within the application software. This general use of the term could also be interchanged with the terms "Function Diagrams" and "Function Diagram Group Modules" as used in the TXS topical report.⁹

SIVAT is used to perform an additional layer of application software validation testing that falls between Function Block testing (component testing) and the FAT, which serves as the system integration and acceptance testing.

Also, see response to RAI 73.

9 Letter from Ronnie L. Gardner (AREVA NP Inc.) to Document Control Desk (NRC), "Request for Additional Information Regarding ANP-10272, 'Software Program Manual for TELEPERM XSTM Safety Systems Topical Report' (TAC No. MD3971)," NRC: 07:020, on May 22, 2007

RAI-86 Question: In determining that the Software Verification and Validation Plan is appropriate to the scale and complexity of the project, does the design technical manager serve as a consultant to the QA manager, or does the design technical manager have authority in that determination?

NRC Concern: The QA and V&V organizations are the only organizations with any authority over the Verification and Validation Plan. The design technical manager is not independent, and could be influenced by non-quality factors such as cost or scheduling issues. The V&V and QA organizations can, however, request the consultation of other organizations and managers, such as the design technical manager, in collecting information from which to base their determination. If this is the intent of the SPM language, it should be modified to clearly state that relationship. The current language of the SPM in this regard is inappropriate.

This raises similar concerns as expressed in RAI #76.

Applicant Reference:

Organization (Page 6-2)

The design technical manager and the QA organization are responsible for determining that the Software Verification and Validation Plan is appropriate to the scale and complexity of the project.

AREVA NP Response to RAI 86: The sentence in Section 6.2.1 will be revised as follows:

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

RAI-87 Question: How does a commonly understood notation facilitate verification of function diagrams? What, precisely, is meant by “a commonly understood notation?”

NRC Concern: The SPM does not explain what notation is used to facilitate verification of function diagrams, why that notation is considered to be commonly understood, or how the use of common notation facilitates the verification process.

While using a commonly understood notation is generally good practice, AREVA NP needs to further explain this statement in the SPM. The commonly understood notation methodology used by AREVA NP should be explicitly stated, as the argument will be self-evident if the identified notation is commonly recognized. If, however, the notation is specialized but commonly understood among AREVA NP employees, for example, the statement should be qualified in such a manner.

Applicant Reference:

Simulation Testing (Page 6-8)

4. *The verification of the function diagrams by the engineers is facilitated by the use of a commonly understood notation.*

AREVA NP Response to RAI 87: The Software Program Manual will be revised as follows:

Verification of the SPACE Function Diagrams by the Verification and Validation engineers 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 diagrams. This check is equivalent to code verification in other code development systems. The code generation verification checks performed by the SPACE tool can be readily verified.

It should be noted that characterization as ‘commonly understood notation’ was taken without attribution from pages 31 and 32 of the TXS Topical Report. The source paragraph stated:

The complete specification captures functional aspects and the system's detailed hardware structure. Nonfunctional aspects such as independence constraints, fault tolerance, and timing requirements are also implicitly contained in the specification. The specification can be prepared by I&C engineers using notations and methodologies that have been common practice in the I&C community. The software specification remains independent of the specific details of the target system. The verification of the specification by the process engineers who prepared the system requirement specification is facilitated by the use of a commonly understood notation.