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

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

RAI 1) Question Summary: Explain why the RETRANS verification tool that is described in the TELEPERM XS Topical Report (TXS TR) is not described in the Software Program Manual (SPM).

Full Text: It is basically understood that the use of the RETRANS tool allowed the code generator to be accepted without exaggerated quality verification demands. For example, TXS TR Section 2.4.3.3.3 says: “As a diverse measure to detect potential software faults not found by the means described in Section 3.2.1, the verification tool "RETRANS" developed by GRS-ISTec is used as an independent testing tool. The generated code can be analyzed by **RETRANS** to identify the function block modules and reveal the connections between them. The result of this process should yield the information elements contained in the design database as input on the SPACE editor for engineering the instrumentation and controls (I&C) functions. A comparison of the result of the validator analysis with the content of the design database for the I&C functions confirms correct application of the tool for code generation and **relieves the code generator of exaggerated quality verification demands** ...”

AREVA NP Response to RAI 1: The last phrase of the quote contained in the full text portion of RAI 1 was omitted. The complete sentence reads as follows:

> A comparison of the result of the validator analysis with the content of the design database for the I&C functions confirms correct application of the tool for code generation and relieves the code generator of exaggerated quality verification demands, **particularly in the introductory phase**. (emphasis added for omitted phrase)

The RETRANS tool is no longer used for the validation of TELEPERM XS (TXS) application software. AREVA NP now validates the generated code on a test bed with a TXS simulation tool referred to as SIVAT (Simulation-based Validation Tool). SIVAT was developed and is maintained by the GRS-ISTec Institute for Safety Technology, Garching, Germany. GRS-ISTec applied it for the purpose of independent verification of TXS applications software in the course of various TXS I&C projects. RETRANS was used 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 code generation process has been proven by a long series of successful TXS I&C installations. Any updates performed to the code generation tool were subject to extensive testing and independent assessment by GRS-ISTec. For the last years of use, RETRANS code verification did not reveal any new findings. For this reason, the total RETRANS verification is no longer performed for current TXS I&C projects and not available on recent software tool releases (release 3.0.x). Those checks that were significant for design verification in the engineering process have been implemented in the scope of an AREVA NP-owned TXS tool. Namely, the automated crosschecking by the *rediff* tool for detecting any deviations between the specified function diagrams assigned to the redundant protection functions (logics, parameters).

A detailed description of SIVAT and its application for verification and validation 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.

Section 2.0 of AREVA NP Report No. NGLP/2004/en/0094 the document provides the description of the tool and its use in the TXS design process. Section 2.4.7 provides an example of the methodology for testing a typical function including sample output data. 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 verification and validation tool for TXS application software.

AREVA NP Report No. NGLP/2004/en/0094 is available for NRC review and audit at an AREVA NP facility.

RAI 2) Question Summary: Explain how the application software development process described in the SPM relates to the one described in the TXS TR.

Full Text: The description in the TXS TR and associated safety evaluation report describe tools and procedural requirements that do not appear to be identified in the SPM. Does the SPM replace, modify, or augment and supplement the application software development process described in the TXS TR? Please explain.

AREVA NP Response to RAI 2: The software life cycle planning process described in section 3 of AREVA NP topical report EMF-2110(NP), TELEPERM XS: A Digital Reactor Protection System, Revision 1, (referred to as 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. Topical report ANP-10272, “Software Program Manual for TELEPERM XS™ Safety Systems”, applies to the development of TXS application software for U.S. projects. The program described in topical report ANP-10272 will be used to address plant-specific actions items 2 and 17 from the NRC Safety Evaluation Report for the TXS topical report issued on May 5, 2000. ANP-10272 augments the TXS topical report by addressing the development of TXS application software for U.S. projects.

RAI 3) Question Summary: Describe how the SPM terms “logic diagrams,” “Functional Requirements Specification,” “Software Requirements Specification,” and “Software Design Description” relate to terms already defined in the TXS TR (i.e. Function Diagram & Software Specification).

Full Text: Does the SPM “Software Requirements Specification” correspond to the TXS TR “Software Specification?” For example, in Section 9.1.3, it says: “The SDD [Software Design Description] uses functional blocks similar to the SPACE tool database to translate the requirements from the SRS [Software Requirements Specification] into logic diagrams. These logic diagrams form the basis of the software logic. The logic diagrams show the inputs, how those inputs are manipulated, and the resulting outputs. Once completed, these diagrams are redrawn in the SPACE tool to generate the code." Therefore are the logic diagrams on the SPM the same thing as the function block diagrams in the TXS TR?

AREVA NP Response to RAI 3: The term "logic diagrams" in section 9.1.3 of ANP-10272 is used to describe the drawings contained the Software Design Description. The information contained in these drawings is then manually entered into the Specification and Coding Environment (SPACE) tool to create the "Functional Diagrams" as discussed in the TXS topical report.

The term "Functional Requirements Specification" in ANP-10272 directly correlates to the term "Vendor/Customer Specification," as described in Section 2.2.2.1 of the NRC safety evaluation report (SER) for the TXS topical report.

The term "Software Requirements Specification" in ANP-10272 directly correlates to the term "Software Requirements Specification," as described in Section 2.2.2.7 in the NRC safety evaluation report (SER) for the TXS topical report. The Software Requirements Specification, as discussed in ANP-10272, follows the guidance of IEEE Std 830, "Recommended Practice for Software Requirements Specifications," and is used to define the requirements for the application software.

The term "Software Design Description" in the SPM directly correlates to the term "Software Design Description" described in Section 2.2.2.9 in the NRC safety evaluation report (SER) for the TXS TR. The Software Design Description as discussed in ANP-10272 utilizes the documentation tool FunBase. This tool merely provides a structure for documenting the Application Software Design.

RAI 4) Question Summary: Describe how the SPM terms "component," "software modules," and "subsystems," relate to terms already defined in the TXS TR (i.e. Function Block, Function Diagram, & Function Diagram Group Modules).

Full Text: The TXS TR uses several terms to describe software pieces and assemblies of software pieces: Function Block modules (FB-module), Function Diagram modules (FD-modules), Function Diagram Group modules (FDG-modules), and application. It appears that the term "component" is used in the TXS TR to refer mostly to system software items and hardware items. In terms of the application software, it appears that the smallest indivisible thing is a FB-module. An application can consist of up to two FDG-modules that, in turn, can consist of FD-modules that, in turn consist of FB-modules. A FB-module is referring to the implementation of a function block - a box on a function diagram. A FD-module can be thought of as the implementation of a function diagram (consisting of one or more pages). The term "application" is used to refer to what is loaded on one function processor (excluding system software). In UML 1.1, a component represents implementation items, such as files and executables. In UML 2, components are considered autonomous, encapsulated units within a system or subsystem that provide one or more interfaces. So a component is probably not an application. In Section 6.0 it says: "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. ... The AREVA NP approach to component testing (called simulation testing) is discussed in section 6.2.7.4.1." However, Section 6.2.7.4.1 does not mention "component." In addition, in Section 6.2.4 it says: "When differing software integrity levels are assigned within the project, the Software Verification and Validation Plan documents the Safety Integrity Level assignment to individual software components, such as requirements, detailed functions, software modules, subsystems ..." How are "requirements" a form of "individual software components?" Therefore, it is not clear what software piece or assembly of software pieces is being referred to as a component.

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.

RAI 5) Question Summary: Describe how the SPM addresses the documentation of the specific design basis of each safety system.

Full Text: IEEE Std 603-1991 Section 4 requires that the specific design basis of a safety system be established. However, the SPM does not describe where this design basis is documented. For example, Section 9.1.4 says: “Each identifiable requirement in the SRS is traceable backwards to the system requirements and either the design bases or regulatory requirements that it satisfies.” Therefore each feature is checked to ensure that it is required (i.e. no extra functionality), but where is it checked that all requirements are implemented in the SRS?

AREVA NP Response to RAI 5: 10 CFR 50.55a(h) requires compliance with IEEE-603-1991, “Criteria for Safety Systems in Nuclear Power Plants,” for digital protection systems. Compliance with the requirements in IEEE 603-1991 is established through the standard design control process for safety-related work, as required by 10 CFR Part 50 Appendix B Criterion III (Design Control). In general, the functional requirements for the safety-related digital protection and control systems are drawn from the Final Safety Analysis Report for operating plants and the Design Control Document for certified designs. This information is generally contained in logic and control diagrams in these licensing basis documents. These functional requirements are an input to the Functional Requirements Specification and then the Software Requirements Specification. The software development process described in ANP-10272 addresses the translation of the functional safety requirements into the final application software. The software development process builds off of the standard design control process; it does not replicate or replace that process.

ANP-10272 Section 1.2 identifies that the Software Requirements Specification is developed during the basic design phase of a project. That section also notes that if the Functional Requirements Specifications is not provided by the customer, AREVA NP will develop it during the basic design phase. Section 9.1.2 describes the documentation requirements for the Software Requirements Specification. Section 9.1.4 describes the documentation requirements for the requirements traceability matrix. Sections 3.4.1 and 6.2.7.1 describe the verification and validation activities for the Software Requirements Specification. Sections 4.3.2 and 6.2.7.3 describe the verification and validation activities for the requirements traceability matrix.

RAI 6) Question Summary: Describe how AREVA intends to address conformance to the standard review plan (SRP).

Full Text 10 CFR 50.34(h) requires an evaluation against the SRP. Therefore each licensee must provide an evaluation of the software development plans against NUREG-0800 Chapter 7, Branch Technical Position (BTP) No. 14. The SPM is an appropriate document to address conformance with the SRP.

AREVA NP Response to RAI 6: 10 CFR 50.34(h) applies to applicants for new plants. As noted in section 1.0 of the topical report, AREVA NP intends to use the Software Program Manual to support digital safety instrumentation and control (I&C) system upgrades at operating nuclear plants and digital safety systems for new nuclear plants. AREVA NP intends to address 10 CFR 50.34(h) in the Design Control Document for the U.S. EPR.

AREVA NP evaluated ANP-10272 and other internal work processes and procedures against BTP HICB-14 as part of the development of the topical report. BTP HICB-14 was the version that was in effect at the time ANP-10272 was submitted to NRC for review. The results of conformance assessment are documented in AREVA NP document 51-9047411-000, "Alignment of the TXS System Application Software Program, as described in the Software Program Manual, with Branch Technical Position HICB-14."

The following is a list of the differences noted in the assessment:

1. From BTP HICB-14, 3.1a, Software Management Plan, Management Characteristics, "The plan should provide…an overview of the system within which the software will reside" – As a generic software program plan, ANP-10272 does not provide an overview of any system within which the software will reside except to say that it will include inputs, processors and outputs. It does provide an overview of the project phases that software development fits into. Descriptions of specific system architecture (using the TXS equipment modules) and system functionality (using TXS application software) are developed for the individual projects.

2. From BTP HICB -14, 3.1a, Software Management Plan, Implementation Characteristics, "It [the Software Management Plan] should describe the approach to be followed for recording the rationale for key decisions made in specifying, designing, implementing, procuring and assessing the software." - This step is not applicable to individual projects. The software generation methodology is the TXS technology that has been approved by NRC.

3. From BTP HICB -14, 3.1c, Software Quality Assurance Plan, Management Characteristics, "*Organization* requires a description of the software QA organization. The plan should describe the boundaries between the software QA organization and other company organizations. Reporting channels should be described." – AREVA NP does not use a separate Software Quality Assurance (QA) organization. The AREVA NP QA organization provides that function as described in ANP-10272 Section 3 and Operating Instruction for the TXS Software Quality Assurance Plan.

4. From BTP HICB -14, 3.1d, Software Integration Plan, Management Characteristics, "The management characteristics that the software integration plan should exhibit include purpose, organization and responsibilities." – No separate software integration organization is required for individual projects. Software integration for the TXS technology has been approved by NRC.

5. From BTP HICB -14, 3.1e, Software Installation Plan, "The software installation plan should exhibit the management, implementation, and resource characteristics listed below." - Software installation is addressed in ANP-10272 Section 2. No separate software installation plan is required for individual projects, since all software is part of the integrated TXS technology.

6. From BTP HICB -14, 3.1f, Software Maintenance Plan, Management, Implementation and Resource Characteristics. Software maintenance is described in ANP-10272 Section 7 and the Operating Instruction for the Software Operations

and Maintenance Plan. These documents describe AREVA NP responsibilities and define interfaces with customer responsibilities.

7. From BTP HICB -14, 3.1h, Software Operations Plan, Management, Implementation and Resource Characteristics. Software operations are described in ANP-10272 Section 7 and the Operating Instruction for the Software Operations and Maintenance Plan. These documents describe AREVA NP responsibilities and define interfaces with customer responsibilities.

8. From BTP HICB -14, 3.1h, Software Operations Plan, Management Characteristics, "*Security* requires a description of the security requirements for operating the software system. The operations plan should identify the controls needed over operation activities to prevent unauthorized changes to hardware, software and system parameters, the monitoring activities needed to detect penetration or attempted penetration of the system, and contingency plans needed to ensure appropriate response to penetration." AREVA NP security responsibilities are described in the Operating Instruction for Cyber Security. Customer security responsibilities will be described in their documents that address pending rule change 10 CFR 73.55(m), "Digital computer and communication networks", and Nuclear Energy Institute document NEI 04-04, "Standard Cyber Security Program for Operating Reactors."

9. From BTP HICB -14, 3.1i, Software Safety Plan, Management Characteristics. "*Organization* requires a description of the software safety organization." Software safety is addressed in ANP-10272 Section 4 and the Operating Instruction for the Software Safety Plan. AREVA NP does not use a software safety organization nor does it perform a specific analysis of the application software to detect hazards. TXS application software is generated by the SPACE tool. 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 failure modes and effects analysis (FMEA), response time analysis, and factory acceptance test (FAT) are sufficient to ensure that there are no software hazards.

10. From BTP HICB -14, 3.1i, Software Safety Plan, Management Characteristics, " A designated safety officer should have clear authority for enforcing safety requirements in the Software Requirements Specification, the design, and the implementation of the software." The Software Supervisor is responsible for the execution of all of the tasks in the Software Safety Plan. As described in ANP-10272 Section 4.3, several parts of the organization are involved in generating the different analyses.

11. From BTP HICB -14, 3.1j, Software Verification and Validation Plan, Management Characteristics, "*Risks* requires a specification of the methods used to identify and manage risks associated with the verification and validation (V&V) process. The plan should specify a method for evaluating the risk to safety associated with each software item. It should describe a method for identifying the risk associated with each V&V task. A contingency plan should be included to identify risk factors that may cause the V&V task to fail to perform its functions, and to recover from any such

failure." Software verification and validation is addressed in ANP-10272 Section 6. The use of the TXS object-oriented automated code generation tools minimizes the inherent risk in the development of the application software.

12. From BTP HICB -14, 3.1j, Software Verification and Validation Plan, Implementation Characteristics, "Regulatory Guide 1.171, "Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants," which endorses ANSI/IEEE Std. 1008, "IEEE Standard for Software Unit Testing," describes acceptable methods for performing unit tests." ANP-10272 Sections 4.3.6 and 9.2.1 address software simulation testing with SIVAT and note that it is planned and executed in accordance with procedures following the applicable recommendations of IEEE 1008, which is endorsed by Regulatory Guide 1.171. The SIVAT testing and its results confirm that the software design is consistent with a basis from the safety analysis. The SPACE function diagrams are used to automatically generate the software. 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. The verification and validation team may also perform independent tests on the software (using SIVAT) and on the integrated system in accordance with the verification and validation plan.

13. From BTP HICB -14, 3.1k, Software Configuration Management Plan, Implementation Characteristics, "The plan should describe procedures to control vendors supplying safety system software." Software vendor controls are described in ANP-10272 Section 3.10. AREVA NP GmbH developed the TXS system software and implemented an approved software QA program for the life cycle of the TXS software. AREVA NP GmbH is an approved supplier for AREVA NP Inc. Software in the TXS system software package is uniquely identified and is subjected to an incoming inspection and is base-lined for configuration control. No other safety-related software (Safety Integrity Level -4) is required to be procured during the software life cycle of TXS projects at AREVA NP. Additional third party non-safety software that is rated Safety Integrity Level-3 or Safety Integrity Level-2 may have to be developed to support the projects. Contracts with any third party supplier of software include the provisions of the Software Quality Assurance Plan.

14. From BTP HICB -14, 3.1k, Software Configuration Management Plan, Resource Characteristics, "The plan should specify a process for selecting configuration management tools." This step is not applicable to individual projects. The configuration management tools for the TXS technology have already been selected.

The differences noted in the assessment are related to two aspects of the AREVA NP application software development process that are not envisioned in the general guidance expressed in BTP HICB-14. The TXS technology is a mature and fully integrated nuclear safety system. As such, certain aspects of a general software development process are not applicable to the individual projects that use the TXS technology. The use of the TXS object–oriented automated code generation tools minimizes the inherent risk in the development of the application software as well as minimizes the potential for human error. These tools support the development of high quality software with a less complex process.

AREVA NP document 51-9047411-000 is available for NRC review and audit at an AREVA NP facility.

RAI 7) Question Summary: Describe the requirements on the Functional Requirements Specification (FRS).

Full Text: Section 9.1.1, "Functional Requirements Specification," does not place any requirements on the form or content of the FRS, nor does it identify any guidance that is followed in producing this document. However, Section 2.2.2.4 of the TXS SE has identified associated procedural requirements that may be applicable to the FRS: '... FAW-3.4, "Contents and Structure of System Specifications for Software Components" ...'

AREVA NP Response to RAI 7: The Functional Requirements Specification referred to in Section 9.1.1 of ANP-10272 is generally a client-supplied document. In some cases the documentation can be supplied by the client's architect/engineering firm. The AREVA NP approach to this topic is predicated on two important foundation elements: the standard design control process for safety-related work, as required by 10 CFR Part 50 Appendix B Criterion III (Design Control) and the use of the NRC-approved TXS object–oriented automated code generation tools for the development of the application software.

For example, the functional requirements for the safety-related digital protection system are based on the information presented in Chapter 7 of Final Safety Analysis Report for operating plants and the Design Control Document for certified designs. This information is generally contained in the logic and control diagrams provided in these licensing basis documents. These functional requirements are an input to the Software Requirements Specification. The name of the Functional Requirements Specification document can vary between clients; however, it is always documentation that describes the functions that the system being designed must accomplish.

It should be noted that the full text of the question mixes two processes. The TXS topical report describes the AREVA GmbH procedures described that were used for the development of TXS operating system software and Function Block library. ANP-10272 addresses the development of project-specific application software using the TXS software development tools. The AREVA GmbH procedures do not apply to the development of TXS application software for U.S. projects.

RAI 8) Question Summary: Describe the requirements associated with the Software Design Descriptions (SDD).

Full Text: Section 9.1.3, "Software Design Description," describes only one aspect of a SDD, the function block/logic diagrams. The SDD must contain material other than just block/logic diagrams. For example: 1) Additional requirements on the SDD are contained in Section 9.4.2, “Coding Standards,” and in Section 9.4.3, “Logic Structure Standards.”; and 2) Section 2.2.2.4 of the TXS SE says: “... FAW-3.5, "Contents and Structure of Design Documents for Software Components,"... FAW-3.5 describes the process by which the software specification is translated into the SDD. FAW-3.6 describes the process by which the SDD is implemented. ...”

AREVA NP Response to RAI 8: The Software Design Description, as described in ANP-10272, contains several types of materials that define and describe the application software and overall system design as it relates to the application software. In addition to the logic diagrams discussed in ANP-10272, the Software Design Description contains descriptions of the overall system design, system architecture, high level functional descriptions (including high level functional logic diagrams), as well as, detailed logic diagrams, task descriptions, input/output signals (per function) and other design/criteria for each application software component. The detailed logic diagrams and associated information are used by the software design team to manually enter the design into the SPACE Engineering Tool.

RAI 9) Question Summary: Describe the conventions for documenting requirements.

Full Text: Section 9.1.2 identifies IEEE 830 as providing guidance for the SRS. This standard contains guidance on broadly accepted practices in requirements documentation. The guidance can be applied to any document that contains requirements, for example the SPM. However it is not clear what standards or conventions are followed in the SPM and associated plans, with respect to requirements documentation. For example: 1) The SPM does not contain a single “shall.”; and 2) The SPM does contain twelve (12) “must’s.” Are there twelve (12) requirements in the SPM?

AREVA NP Response to RAI 9: ANP-10272 is AREVA NP’s Software Program Manual for TXS application software development. As such, it is the upper tier requirements document. The statements in ANP-10272 are considered to be the programmatic requirements necessary to implement the TXS application software program.

AREVA NP uses the term “shall” to denote requirement statements in Operating Instructions and project-specific plan documents. The AREVA NP Procedures and Policies Dictionary defines shall as “Denotes a requirement.” The term “shall” is used in the Operating Instructions and project-specific plans to implement the ANP-10272 requirements.

IEEE Std 830-1993 is a recommended practice for Software Requirements Specifications. The assertion in the RAI that the guidance in the IEEE Std 830-1993 can be applied to another document that contains requirements may be true; however, it is not correct for its use at AREVA NP. With regard to ANP-10272, IEEE Std 830 - 1993 applies exclusively to the Software Requirements Specification. At AREVA NP the Software Requirements Specification is a specific document that extracts functional requirements for the software from the Functional Requirements Specification.

RAI 10) Question Summary: Describe how the Failure Modes and Effects Analysis (FMEA), as described in Section 4.3.3 of the SPM, will follow the guidance of IEEE 379-2000.

Full Text: It is not clear if AREVA means that the FMEA document shall conform to all of the requirements in IEEE 379-2000. Will the FMEA also follow the recommendations and permissions in IEEE 379-2000. Note: IEEE 379 refers to IEEE 352 for reliability analysis. IEEE 352 contains guidance for FMEAs. However, Section 4.3.3 says: "The FMEA follows the guidance of IEEE 379..."

AREVA NP Response to RAI 10: AREVA NP will meet the requirements (*shall* statements) of IEEE 379-2000, "Application of the Single-Failure Criterion to Nuclear Power Generating Station Safety Systems." to establish conformance with the requirements of IEEE Std. 603-1991, specifically the single-failure criterion as stated in clause 5.1. The guidance in section 4.1 of IEEE 352-1987, "Guide for General Principles of Reliability Analysis of Nuclear Power Generating Station Safety Systems," will be used for the FMEA analyses for TXS projects, unless customer requirements specify a different format.

RAI 11) Question Summary: Describe the difference in meaning of the various conformance claims.

Full Text: The SPM claims conformance to certain criteria in different ways, and it is not clear whether these difference in terminology reflect differences in meaning (if so, how), or simply are different for linguistic diversity and readability reasons. For example

1) Section 3.1 says: "The Software Quality Assurance Plan fulfills the requirements for a software quality assurance plan in accordance with IEEE 730 ... 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." This statement seems to say both: a) that the SQAP conforms to IEEE 730; and b) that it does not do so completely. Please explain.

2) Section 9.1.2 says: "The SRS is written following the content and format recommendations of IEEE 830, which is endorsed by Regulatory Guide 1.172." Therefore it is understood that Section 9.1.2 requires that each "shall" and "must" in the modified standard is followed. It would be more apparent to say that the SRS shall conform to the guidance in RG 1.172.

AREVA NP Response to RAI 11: The topical report makes two statements about the software quality assurance plan requirements described in IEEE 730-2002, "Standard for Software Quality Assurance Plans." First, the topical report states that the AREVA NP Software Quality Assurance Plan, as described in the topical report and the associated Operating Instruction, is the software quality assurance plan required by IEEE 730. Some of the activities described in the Software Quality Assurance Plan are performed by the independent AREVA NP (QA) organization. The governing control document for activities performed by the QA organization is the AREVA NP Quality Management Manual. This information is not duplicated in the Software Quality Assurance Plan.

The intent of ANP-10272 section 9.1.2 is that the Software Requirements Specifications will conform to IEEE Standard 830-1993, as augmented by Regulatory Guide 1.172, “Software Requirements Specifications for Digital Computer Software Used In Safety Systems of Nuclear Power Plants” dated September 1997.

RAI 12) Question Summary: Describe how the logic diagrams in the SDD are redrawn.

Full Text: Section 9.1.3 says that the logic diagrams in the SDD are redrawn in the SPACE tool. Is this redrawing done by a human being, or is it done by a software tool?

AREVA NP Response to RAI 12: The detailed logic diagrams and associated information contained the Software Design Description are used by the software design team to manually enter the design into the SPACE Engineering Tool.

RAI 13) Question Summary: Describe the project specific plans that will be created for each project.

Full Text: The SPM and associated plans describe a program that is augmented and supplemented with project specific plans. However it is not clear what specific plans are required to augment and supplement the SPM and associated plans in order to address all of the NRC requirements and guidance. Nor is it clear what is expected to be in each project specific plan. The use of the term “plan” to describe programmatic aspects and also project specific aspects is confusing. The fact that both programmatic and project specific plans exist for software configuration management plan (SCMP) and software verification and validation plan (SVVP) make it hard to determine, when only SCMP or SVVP is used, which one is being referred to. It is suggested that programmatic documents use the term “program” and that project specific documents use the word “plan.” For example, the following section says:

> ABSTRACT: “The combination of the Software Program Manual and the five plans listed above, which are implemented in AREVA NP operating instructions, constitute a program ...”
>
> Section 5.1.2: “A project plan or specific Software Configuration Management Plan would cover ...”
>
> Section 5.2.2: “...Any special agreed upon requirements are incorporated into the project plan ... These configuration status accounting reports are published periodically at the frequency established in the project plan. ... and document this verification and validation activity in accordance with the project-specific verification and validation plan.”

AREVA NP Response to RAI 13: The hierarchy of the documents that cover the Application Software Development at AREVA NP is as follows:

1. Software Program Manual (Upper Tier Programmatic Guidance)

2. Operating Instructions for the Software Quality Assurance Plan, Software Safety Plan, Software Configuration Management Plan, Software Verification and

Validation Plan, Software Operation and Maintenance Plan, and other topics (Detailed Generic Implementation Guidance)

3. Project-specific plans for the Software Quality Assurance Plan, Software Safety Plan, Software Configuration Management Plan, Software Verification and Validation Plan, Software Operation and Maintenance Plan (Additional Project-Specific Implementation Guidance)

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

In addition, a document called the Project Plan is developed for each project. The Project Plan is a project management document that details topics such as the project controls plan, the overall project quality plan, the project human resources plan, the project communications plan, the project risk management plan, the project procurement/contract management plan, and the project close out plan.

RAI 14) Question Summary: The requirements to identify changes between revision of documents are not addressed in the SPM. Describe the requirements to identify changes between two revisions of a document.

AREVA NP Response to RAI 14: The AREVA NP Quality Management Manual and associated administrative procedures describe document control requirements. These controls specify that revisions to documents be described in the Records of Revisions page included in each document.

The code within the SPACE Engineering tool is controlled via the Software Configuration Management Plan. Only authorized and documented changes are allowed. First, the application Software Design Description document is revised to reflect any changes. Then, changes are made to the SPACE database. Changes to function diagrams and connections between function diagrams are tracked via date and time stamps within the SPACE engineering tool.

RAI 15) Question Summary: Describe the software quality metrics used.

Full Text: The SPM makes reference to “software quality metrics” but does not describe which specific metrics will be used. For example, Section 5.3.9 says: “The open item tracking system is also the primary source of statistical information for software quality metrics.”

AREVA NP Response to RAI 15: See ANP-10272 section 6.3 and the response to RAI 17 for a more complete description of software quality metrics.

RAI 16) Question Summary: Describe what “operating instructions” are and how they are used.

Full Text: It appears from the use of the term “operating instructions” in the SPM, that they can be like Quality Assurance (QA) procedures (i.e. programmatic in nature) and at times that they can be project specific. However it is not clear how or when an operating instruction is programmatic and when it is project specific.

AREVA NP Response to RAI 16: Within the AREVA NP system, Operating Instructions are formal documents that specify or describe how an activity is to be performed within a department with no actions required by other departments. Operating Instructions may implement policies, administrative procedures, and QA Program requirements. Operating Instructions define generic programmatic requirements.

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

RAI 17) Question Summary: Describe why the specific metrics are used?

Full Text: Typically metrics are part of a metrics program, and have relatively little value when first applied. It is only through programmatic development that metrics have value. For example, it is not obvious why the following metrics measure the effectiveness of Verification and Validation (V&V): 1) “History of project deliverables compared to schedule commitments,”; 2) “Total number of verification and validation open items in the open item backlog as a function of calendar time,”; and 3) “Length of time to close a verification and validation open item after identification.” For example is it good or bad to have a lot of V&V open items in the backlog? (Good: V&V is finding stuff faster than design can fix it; Bad: V&V is not closing the items after design has fixed them or V&V is not working with design)

AREVA NP Response to RAI 17: AREVA NP has re-evaluated the metrics listed in ANP-10272 section 6.3.2 as measures of the effectiveness of the verification and validation activities, as noted below:

- **History of project deliverables compared to schedule commitments** - It has been concluded that this metric does not measure the effectiveness of the verification and validation activities.

- **Total number of verification and validation open items in the open item backlog as a function of calendar time** - For any project, the effectiveness of the verification and validation process can be measured by the decrease of the number open items identified as calendar time goes by, especially towards the end of the project. This metric also provides an indication of the amount of effort still required to complete the project. AREVA NP believes this metric is useful in determining the effectiveness of the verification and validation activities.

- **Number of project open items discovered by verification and validation compared to the total number of open items -** The verification and validation program should be identifying a comparable number of open items as the design and testing organizations, unless the design process is very mature and producing essentially error-free work. AREVA NP believes this metric is useful in determining the effectiveness of the verification and validation activities.

- **Severity and risk statistics associated with errors and open items discovered during verification and validation activities -** The severity level of open items is an indicator of the nature of the errors being made and the integrity of key design process controls. AREVA NP believes this metric is useful in determining the effectiveness of the verification and validation activities.

- **Length of time taken to close a verification and validation open item after identification -** It has been concluded that this metric does not measure the effectiveness of the verification and validation activities.

- **Stability of the software requirements traceability matrix requirements statements based on the number of revisions made -** It has been concluded that this metric does not measure the effectiveness of the verification and validation activities.

- **Number of technical comments made to draft design output documents and an assessment of whether any of the comments were previously considered by the independent reviewer -** It has been concluded drawn that this metric does not measure the effectiveness of the verification and validation activities.

- **Number of test anomalies discovered during independent verification and validation testing -** This metric provides an additional insight beyond tracking the total number of open items found in all activities. The quantity of anomalies that have made it through the verification process and are found in the tested software provides an indication of the effectiveness of the verification activities. AREVA NP believes this metric is useful in determining the effectiveness of the verification and validation activities.

- **Verification and validation coverage, that is the fraction of the software product reviewed by verification and validation -** It has been concluded drawn that this metric does not measure the effectiveness of the verification and validation activities.

- **Verification and validation man-hours charged to the project in each phase of the development life cycle -** It has been concluded that this metric does not measure the effectiveness of the verification and validation activities.

RAI 18) Question Summary: Describe the meaning of "project."

Full Text: When reference is made to project specific plans, it is not clear if there is a set of plans produced for each contract (i.e. a new plant), a set of plans is produced for each system (i.e. engineered safety features (ESF)), or a layers of plans at the project and system levels are produced.

AREVA NP Response to RAI 18: Within the AREVA NP system, projects are organized to support both integrated engineering as well as financial and contract management. Two general organization schemes are envisioned: operating reactor retrofit projects for multi-site reactors that would be subdivided by unit and new plant certification projects that would be subdivided by major systems (e.g., protections system, safety control system, and non-safety control systems).

Project-specific plans are developed to be sent to a client as a document deliverable that defines the programmatic requirements used for a project activity, when the client requires such documentation. Project-specific plans incorporate the generic program requirements from the corresponding Operating Instruction in effect at that time. Project-specific plans can also augment the procedure requirements established in the generic procedures, as specified in customer requirements or unique system requirements, or take some exception to a provision of the general plan, when justified. In the first example a single set of project-specific plans would be used for all the units of a multi-unit project at a single site. In the second example, project-specific plans might be used to address unique requirements for a non-safety control system that has an interface connection with a safety-related system.

RAI 19) Question Summary: Describe the difference in meaning between the following terms: safety goals, software risk, software hazard, and software error.

Full Text: From the way that these terms are used by Areva, it is not clear what distinction Areva is making between the meaning of these terms. It should be noted that proper operation in accordance with design, and safety are generally considered two distinct concepts. For example, a revolver is a highly reliable piece of equipment, but is questionably safe. In addition, safety is a system issue, not strictly a software issue. Therefore ensuring that the software functions as designed, and ensuring that the system is safe are two distinct types of evaluations. Of course, errors in the implementation of a design can be unsafe, but this is not the only way that an item can be unsafe. For example, single event effects are one way that faults can be created in a properly programmed system. The proper handling of faults is desirable. However, Section 4.0 says: "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 factory acceptance test (FAT) are sufficient to ensure that there are no software hazards."

AREVA NP Response to RAI 19: The following meanings are ascribed to the terms safety goals, software risk, software hazard, and software error in ANP-10272:

Safety Goal - Provide reasonable assurance that the TXS application software performs its design basis safety function, as defined in 10 CFR 50.2. These design basis functions are described in the Functional Requirement Specification for a project.

Software Risk - A measure that combines both the likelihood that a software hazard will cause a problem and the severity of that problem. This definition is consistent with IEEE Standard 1228-1994, "IEEE Standard for Software Safety Plans." A software risk is minimized in the design process by using the TXS SPACE tool, SIVAT testing, and FAT. The risk is further minimized through the verification and validation process.

Software Hazard – A software design error that could lead to an unintended operation or failure to operate when required. The analyses and tests defined in ANP-10272 section 4.3 are specified to provide reasonable assurance that software hazards are eliminated. This definition is consistent with IEEE Standard 1228-1994.

Software Error - An actual mistake in the software. The TXS technology is a mature and fully integrated nuclear safety system. The development and qualification processes for the TXS operating system software and Function Block library are described in the TXS topical report. ANP-10272 describes the development and qualification processes for TXS application software using the Function Block library. The use of the TXS object–oriented automated code generation tools minimizes the inherent risk in the development of the application software as well as minimizes the potential for human error. These tools support the development of high quality software. Together, the two TXS software processes provide reasonable assurance that software errors are minimized.

All safety functions of a nuclear plant system are identified as design inputs and must be satisfied by the design. As such, the system must be designed, constructed and tested to perform those safety functions. It is no different with systems that contain software. Safety system software is designed to successfully perform the safety function of the system and to ensure that it does not present any conditions that would result in an unsafe configuration of the plant. The design of the TXS application software is first done with logic diagrams that describe how the software is to function. It is then analyzed for any adverse effects functions that might occur as a result of the design. Next, the function diagrams are generated from the logic diagrams in SPACE and the application software is generated. The software automatically generated by the SPACE tool It is then tested using SIVAT to verify that all of the safety functions are properly executed by the software and that no unintended events or states exist. The key analyses of the system design (and software), include the diversity and defense-in-depth analysis, the application software requirements traceability matrix, the failure modes and effects analysis, the response time analysis, the verification and validation reports, SIVAT testing and software test report, and the criticality analysis.

RAI 20) Question Summary: Describe how the software safety plan is implemented in the context of the system safety program.

Full Text: Since the software safety plan "follows the concepts IEEE 1228," and since IEEE 1228 says that the software safety program is implemented within the system safety program, clarification of how the Areva program addresses this aspect is desired.

However, Section 4.0 says: “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 20: The AREVA NP approach to this topic is predicated on two important foundation elements: the standard design control process for safety-related work, as required by 10 CFR Part 50 Appendix B Criterion III (Design Control) and the use of the NRC-approved TXS object–oriented automated code generation tools for the development of the application software. The responsibility to produce a safe application software product is not separate from the responsibility to produce a quality product, or a functional product. There is not a separate organization that is responsible for each of these functions. AREVA NP does not use a software safety organization, based on the use of the standard safety design control process and the software development quality controls. Instead, each of these functions is the responsibility of the project organization as defined in the topical report.

The Project Manager has the following responsibilities in the conduct of the Software Safety Plan Operating Instruction:

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

b. Coordinate safety task planning with other organizational components or functions, such as development, system safety, software quality assurance, software reliability, software configuration management, V&V, and software testing.

c. Obtain, allocate, and monitor resources for effective implementation of the Software Safety Plan.

d. Participate in audits of Software Safety Plan implementation.

e. Coordinate technical issues related to software safety with the AREVA NP Inc. project Lead Software Engineer.

f. Ensure training in methods, tools and techniques used in software safety tasks for the project and V&V personnel. Ensure that the training is documented in accordance with AREVA NP administrative procedures.

g. Communicate any safety concerns in accordance with the AREVA NP Corrective Action Program.

The Software Design Supervisor has the following responsibilities in the conduct of the Software Safety Plan Operating Instruction:

a. Responsible for the development of the Software Safety Plan Operating Instruction and any needed revisions.

b. Responsible for the overall conduct of software safety activities.

c. Technical direction to members of the Software Design Group for software safety activities.

d. Ensures the Software Design Group project specific training in provided and documented in accordance with AREVA NP administrative procedures.

e. Communicate any safety concerns in accordance with the AREVA NP Corrective Action Program.

The standard's requirement to have a separate software safety organization from the software design group is not required at AREVA NP.

An underlying assumption of IEEE 1228-1994 is that there is a group that is doing coding work from some set of functional requirements or diagrams. For TXS application software, the code is generated by TXS object–oriented automated code generation tool (SPACE). As such, the application software design group at AREVA NP does not create code. Instead, the software design group performs the software safety activities as part of their application software development responsibilities.

Additional information on the software safety plan is provided in response to RAI 22.

RAI 21) Question Summary: Clarify Areva's concept of the requirements for defense-in-depth and diversity with respect to manual controls.

Full Text: It is not clear what Areva believes to be required with respect to manual controls and Defense-in-Depth and Diversity. The SRP (NUREG-0800) was last updated in 1997. In 1999 10CFR50.55a(h) requires that safety systems meet the requirements of IEEE 603-1991. IEEE 603 Section 6.2 requires a manual means to initiate each function that is required of the safety system. IEEE Section 6.2.1 requires that the manual means, associated with an automatic means, be implemented in a diverse manner. The implication is that all of these manual means are of the same quality as the automatic means. In addition, Branch Technical Position No. 19 "Guidance for Evaluation of Defense-in-Depth and Diversity in Digital Computer-Based Instrumentation and Control Systems" basically says that if the safety system is implemented as a programmable system, then a Defense-in-Depth and Diversity analysis is required, since programmable systems are vulnerable to common cause failures. This analysis may conclude that a diverse system is also required. Therefore if the safety system manual controls are implemented using software, then a diverse set of manual controls may also be required. However, Section 4.3.1 says: "The diversity and defense-in-depth analysis is performed to assess the adequacy of diversity afforded by the system design, to ensure that adequate defense-in-depth has been provided in the design, and to verify that the displays and manual controls for critical safety functions initiated by operator action are diverse from computer systems used in the automatic portion of the reactor protection and engineered safety features actuation systems."

AREVA NP Response to RAI 21: The question states that section 6.2.1 of IEEE 603-1991 "requires that the manual means, associated with an automatic means, be implemented in a diverse manner." IEEE 603-1991 does not specify any diversity requirements for manual controls, as noted below.

> 6.2.1 Means shall be provided in the control room to implement manual initiation at the division level of the automatically initiated protective actions. The means provided shall minimize the number of discrete operator manipulations and shall depend on the operation of a minimum of equipment consistent with the constraints of 5.6.1.
>
> and
>
> 5.6.1 Between Redundant Portions of a Safety System. Redundant portions of a safety system provided for a safety function shall be independent of and physically separated from each other to the degree necessary to retain the capability to accomplish safety function during and following any design basis event requiring that safety function.

AREVA recognizes that IEEE 603-1991 requires a safety-related set of manual controls. In addition, the guidance in Point 4 of NRC Branch Technical Position (BTP) 7-19, "Guidance for Evaluation of Diversity and Defense-in-Depth in Digital Computer-Based Instrumentation and Control Systems," suggests that required manual controls should be independent and diverse from the computer-based safety systems identified in Points 1 and 3. Standard Review Plan section 7.8, "Diverse Instrumentation And Control Systems," notes that diverse manual controls are not required to be safety-related.

Depending on the design details, safety-related manual controls required by IEEE 603-1991 can also satisfy the BTP 7-19 diversity test and be credited for diverse mitigation capability. The discussion in ANP-10272 section 4.3.1 speaks directly to addressing BTP 7-19 Point 4 and is silent on the IEEE 603-1991 requirements.

RAI 22) Question Summary: Describe the software safety analysis methods, tools and techniques (MTT) that will be used to implement the Software Safety (SS) Plan (SSP), and what work products these MTTs will be applied to.

Full Text: Conceptually, IEEE 1228 requires various software safety analyses to be performed (i.e. SS Requirement Analysis, SS Design Analysis, ...) and that the specific types be identified. In the SSP (SPM Section 4), Areva describes eight (8) safety analysis activities: 1) Diversity and Defense in Depth Analysis; 2) Application Software Requirements Traceability Matrix; 3) Failure Modes and Effects analysis; 4) Response Time Analysis; 5) Verification and Validation Reports; 6) Software Test Report and SIVAT testing; 7) Criticality Analysis; and 8) Factory Acceptance Test Report. However, each of these is addressed by different guidance as is explained below. Therefore it is not clear what activities are required by the SSP (i.e. not required elsewhere). If all activities are performed in accordance with a plan or procedure, what activities would not be performed if the SSP were eliminated?

1) BTP-19 provides guidance to the staff for reviewing a defense-in-depth and diversity analysis.

2) The acceptance criteria for the Software Quality Assurance Plan (SQAP) says that the SQAP should ensure that traceability is maintained throughout all phases, and an RTM is one way of documenting this.

3) IEEE 603 Section 5.15 requires a reliability analysis, and NUREG-0800 Section 7.1-C contains the guidance for reviewing this analysis. An FMEA is considered one way to address this requirement.

4) Response time is a functional or performance requirement, and ensuring that an application responds in the required time period has traditionally not been considered a safety analysis activity.

5) BTP-14 provides acceptance criteria for Software Verification and Validation Plans.

6) Testing can be addressed under SCMP, SVVP, & SQAP.

7) Not much guidance exists regarding graded levels of quality. Basically everything in a safety system should be of high quality. It is not clear how a criticality classification of software is relevant to safety. Where or how is the Safety Integrity Level of a module considered?

8) Testing can be addressed under SCMP, SVVP, & SQAP. However, Section 4.0 says: "The plan follows the concepts of IEEE 1228 but does not fully comply ... AREVA NP does not use a software safety organization nor does it perform a specific analysis of the application software to detect hazards." Please explain.

AREVA NP Response to RAI 22: The AREVA NP approach to this topic is predicated on two important foundation elements: the standard design control process for safety-related work, as required by 10 CFR Part 50 Appendix B Criterion III (Design Control) and the use of the NRC-approved TXS object–oriented automated code generation tools for the development of the application software.

For example, the functional requirements for the safety-related digital protection system are based on the information presented in Chapter 7 of Final Safety Analysis Report for operating plants and the Design Control Document for certified designs. This information is generally contained in the logic and control diagrams provided in these licensing basis documents. These functional requirements are an input to the Software Requirements Specification. The failure modes and effects analysis and response time analysis are examples of design activities that ensure that the single failure requirements are satisfied and safety analysis assumptions are satisfied.

The criticality analysis is an important task that is required by the Software Safety Plan. Clearly, the safety-related software is classified as Safety Integrity Level-4; however, other non-safety software that is part of the project (e.g., gateway or graphical service monitor) must be assessed for impacts from system failures. The criticality analysis is a structured evaluation of the software characteristics for severity of impact of system failure, system degradation, of failure to meet software requirements or system objectives of software developed in accordance with ANP-10272. The criticality analysis document is prepared by the development organization and Safety Integrity

Levels are assigned to all software items that are developed for the project. The Safety Integrity Level is reviewed as part of the verification and validation activities to verify that the assigned Safety Integrity Level is appropriate for the application. Subsequent verification and validation activities are based on the Safety Integrity Level. The criticality analysis would not be performed if the Software Safety Plan was eliminated.

The software development process described in ANP-10272 addresses the translation of the functional safety requirements into the final application software. The software development process builds off of the standard design control process; it does not replicate or replace that process. The use of the TXS object–oriented automated code generation tools minimizes the inherent risk in the development of the application software as well as minimizes the potential for human error. These tools support the development of high quality software with a less complex process. Software testing with SIVAT and the FAT are examples of design and testing activities that are use to ensure proper software development. Application software requirements traceability is part of the verification and validation activities.

The diversity and defense-in-depth analysis is a systematic way to assess the adequacy of diversity afforded by the system design, to ensure that adequate defense-in-depth has been provided in the design, and to verify that the displays and manual controls for critical safety functions initiated by operator action are diverse from computer systems used in the automatic portion of the reactor protection and engineered safety features actuation systems.

RAI 23) Question Summary: Describe how the “consideration” of the extended FMEA, as described in Section 4.3.3 of the SPM, follows the guidance of IEEE 379-2000.

Full Text: IEEE Std 379 does not mention an extended FMEA, and therefore it is not clear how the extended FMEA can follow IEEE 379. It is not clear if AREVA means that the extended FMEA document shall conform to all of the requirements in IEEE 379-2000. Will the extended FMEA follow the Recommendations and permissions in IEEE 379-2000? How will it be documented that “consideration was given?” IEEE 379 refers to IEEE 352 for reliability analysis. However IEEE 352 does not mention an extended FMEA. However, Section 4.3.3 says: ‘On a project to project 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.’ Please explain.

AREVA NP Response to RAI 23:

IEEE 379-2000 suggests in section 5.5 “Common-cause failures,” 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.

As noted in ANP-10272 section 4.3.3, AREVA NP will consider performing a limited analysis of multiple random hardware and software failures on a project to project basis. The guidance in section 4.5, “Extended Qualitative Analysis for Common-Cause Failures,” of IEEE 352-1987 will be used for any extended FMEA analyses performed for TXS projects, unless customer requirements specify a different format. Extended FMEA analyses will not consider the effects of a software common mode failure because this kind of failure is handled by the diversity and defense-in-depth analysis discussed in section 4.3.1 of the topical report. 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 Std. 603-1991. Instead, extended FMEA analyses, when performed, are used to provide additional insights regarding risk, reliability, or other performance objectives specified by the customer.

RAI 24) Question Summary: Describe the relationship between tasks that address the Single Failure analysis and Reliability Analysis requirements.

Full Text: IEEE 603 Section 5.1 contains requirements for the safety system to perform all safety functions in the presence of single failures (therefore an analysis is implied), and Section 5.15 contains requirements for reliability analysis. Both sections refer to IEEE 352 and 577 for guidance on reliability analysis. IEEE 352 describes an FMEA, as one step in a system reliability analysis. Therefore it is not clear if the FMEA is intended to address, in part, both single failure and reliability analysis requirements. For example Section 4.3.3 says: “The FMEA examines the effects of random single failures on the ability of the safety system to perform its required safety functions.” Please explain.

AREVA NP Response to RAI 24:

AREVA NP will meet the requirements (*shall* statements) of IEEE 379-2000, ”Application of the Single-Failure Criterion to Nuclear Power Generating Station Safety Systems,” to establish conformance with the requirements of IEEE Std. 603-1991, specifically the single-failure criterion as stated in clause 5.1. The guidance in section 4.1 of IEEE 352-1987, “Guide for General Principles of Reliability Analysis of Nuclear Power Generating Station Safety Systems,” will be used for the FMEA analyses for TXS projects, unless customer requirements specify a different format.

AREVA NP will meet the requirements IEEE 603-1991 clause 5.15, reliability, for those systems that have either quantitative or qualitative reliability goals established by the customer. Appropriate analysis of the design shall be performed in those cases in order to confirm that such goals have been achieved using IEEE 352-1987 and IEEE 577-1976, “Requirements for Reliability Analysis in the Design and Operation of Safety Systems for Nuclear Power Generating Stations,” as general guidance for the reliability analyses for TXS projects, unless customer requirements specify a different format. The credible failure modes for TXS hardware that are identified through the FEMA process are an input to the evaluation of TXS reliability.

RAI 25) Question Summary: Describe how the Areva software development program addresses the scale (e.g. size and duration) of a project implemented under it.

Full Text: Software development projects can be of many sizes, and durations. It is expected that the importance of certain documentation will vary with differences in size and duration of the project. However, it is not clear what programmatic means exist to address the scale of a project. Partly this issue could be addressed by describing the software development program in the context of the system design and development. For example, a plant-wide digital I&C design would require more documents and more document types than a single system modernization. However this additional documentation may not be considered part of the software development process. The software development program is a part of system development, and it is hard to determine that the software development program is complete without knowing what is addressed in the system development program.

AREVA NP Response to RAI 25: The AREVA NP TXS Application Software development process described in ANP-10272 is used for all projects, regardless of the "scale" of the project. The general flow starts with the customer specification for the system. Next, a Functional Requirements Specification is generated. Then, the Software Requirements Specification is generated based on the Functional Requirements Specification. And then, the Software Design Description is generated based on the Software Requirements Specification. Finally, the software is generated using the SPACE tool. This process is the same for each system developed. In some cases, a project may encompass more than one system. In that case, each system would have its own set of design documents, as described above. Or, two or more systems could be combined into one replacement system that provides all of the functions described in two different specifications. In either case, the software documentation would be organized in the same manner.

See responses to RAIs 1, 18, and 22 for additional information.

RAI 26) Question Summary: Describe how the Areva software development program addresses the different implementation contexts (e.g. design certification vs modernization) of a project implemented under it.

Full Text: The letter that submitted the SPM to the NRC identified that it was intended that the SPM would be referenced by both existing plant modernization projects, and by the US EPR design certification (DC). However the SPM does not describe how these two implementation contexts (i.e. modernization vs DC) are treated differently.

AREVA NP Response to RAI 26: The same software development process controls are applied to modernization projects and a design certification project.

See response to RAI 25 for additional information.

RAI 27) Question Summary: Describe the configuration management process for changing a setpoint in the protection system when it is in operation.

Full Text: It is understood that a setpoint change is essentially a software change. It is not clear how the SCMP addresses these changes.

AREVA NP Response to RAI 27: The responsibility for software configuration management is the responsibility of the customer once the systems is installed and commissioned. Setpoint changes (i.e., changeable parameters as defined in the TXS topical report) can be made by the customer through their design control and software configuration management processes.

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

a. The identification and correction of software errors, performance failures, and/or implementation problems (corrective maintenance).

b. Modifications to permit the software system to run in a different operating environment, with different types of data, or to incorporate new requirements (adaptive maintenance).

c. Modifications to enhance performance, improve cost effectiveness, or otherwise improve the software system (perfective maintenance).

The Software Operations and Maintenance Plan follows the life cycle planning for operations and maintenance guidance of IEEE Standard 1074-1995, “IEEE Standard for Developing Software Life Cycle Processes.”

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

The implementation of adaptive and perfective maintenance changes shall be set up as a new TXS project following the requirements of the AREVA NP Operating Instruction for TXS projects. This type of maintenance requires an approved change order or purchase order and an approved set of design inputs prior to the start of work. The software development process described in ANP-10272 would be used for these projects.

RAI 28) Question Summary: Explain how the SPM addresses the configuration of parameters and definition of trace data that can be set through the runtime environment (RTE) when in the PARAM operating mode.

Full Text: The TXS TR (in Section 3.1.3.4) described four operation modes (OPERATION, PARAM, TEST, & DIAGNOSIS). It is possible to permanently change the functioning of the processor module when not in the OPERATION mode. However it is not clear how the documentation of these changes are addressed in the SCMP. It is expected that setpoint changes will be made using these features, correct?

AREVA NP Response to RAI 28: Entry into processor modes other than OPERATION is controlled via access to the Service Unit and physical access to permission release devices for each processor (i.e. the key switches). Modification of online parameters can only be performed with the processor is in the PARAM mode.

As stated in ANP-10272 section 5.3.8:

> "In the course of making modifications to changeable parameters, the online database is updated to ensure consistency with archived plant documentation, animated function diagrams and current software status. In cases where the online changeable parameter is copied over into the new specification, an open item for software modification is opened and processed following the change control procedures described above.
>
> After a parameter change, the list of changeable parameters has to be regenerated. This list is compared with the online database of the service unit to verify that only the intended parameters were changed."

The changeable parameter list is controlled by the Software Configuration Management Plan. The Software Configuration Management Plan provides additional detail on the tools and methods used to perform this verification.

TEST and DIAGNOSTIC mode allow modifications to the processor operation (e.g., forcing of output or download of new code); however, the processor can only be returned to OPERATION mode with a complete reboot. This reboot would cause the processor to reload the changeable parameters from the EEPROM.