U.S. NUCLEAR REGULATORY COMMISSION
OFFICE OF NUCLEAR REGULATORY RESEARCH

August 1996
Division 1
Draft DG-1058

DRAFT REGULATORY GUIDE

Contact: J. Kramer (301)415-5891

DRAFT REGULATORY GUIDE DG-1058

SOFTWARE REQUIREMENTS SPECIFICATIONS FOR DIGITAL COMPUTER SOFTWARE USED IN SAFETY SYSTEMS OF NUCLEAR POWER PLANTS

A. INTRODUCTION

In 10 CFR Part 50, "Domestic Licensing of Production and Utilization Facilities," paragraph 55a(a)(1) requires, in part,[1] that systems and components be designed, tested, and inspected to quality standards commensurate with the safety function to be performed. Criterion 1, "Quality Standards and Records," of Appendix A, "General Design Criteria for Nuclear Power Plants," to 10 CFR Part 50 requires, in part,[1] that appropriate records of the design and testing of systems and components important to safety be maintained by or under control of the nuclear power unit licensee throughout the life of the unit. Appendix B, "Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants," to 10 CFR Part 50 describes criteria that a quality assurance program for systems and components that prevent or mitigate the consequences of postulated accidents must meet. In particular, besides the systems and components that directly prevent or mitigate the consequences of postulated accidents, the criteria of Appendix B also apply to all activities affecting the

[1] In this draft regulatory guide, many of the regulations have been paraphrased; see 10 CFR Part 50 for the full text.

safety-related functions of such systems and components as designing, purchasing, installing, testing, operating, maintaining, or modifying. A specific requirement is contained in 10 CFR 50.55a(h), which requires that reactor protection systems satisfy the criteria of IEEE Std 279-1971, "Criteria for Protection Systems for Nuclear Power Generating Stations."[2] Paragraph 4.3 of IEEE Std 279-1971[3] states that quality of components is to be achieved through the specification of requirements known to promote high quality, such as requirements for design, inspection, and test.

Several of the General Design Criteria (GDC) of Appendix A, including Criteria 12, 13, 19, 20, 22, 23, 24, 25, and 28, describe functions that are part of the design bases of nuclear power plants and that would be included in the software requirements specification (SRS) of any digital computer software that is part of basic components that perform these functions. In addition to the criteria of Appendix A, Appendix B to 10 CFR Part 50 provides quality assurance criteria that design documentation for nuclear reactor safety systems must meet. Criterion III, "Design Control," requires measures for design documentation and identification and control of design interfaces, as well as measures for verifying or checking the adequacy of the design.

This regulatory guide endorses IEEE Std 830-1993, "IEEE Recommended Practice for Software Requirements Specifications,"[3] as amended below in the Regulatory Position. IEEE Std 830-1993 describes a method acceptable to the NRC staff for complying with the NRC's regulations for achieving high functional reliability and design quality in software used in safety systems.[4] In particular, the method is consistent with GDC 1 and the criteria for quality assurance programs in Appendix B as they apply to the development of software requirements specifications. The criteria of Appendices A and B apply to systems and related quality assurance processes, and if those systems include software, the requirements extend to the software elements.

In general, information provided by regulatory guides is reflected in the Standard Review Plan (NUREG-0800, currently under revision), which is used by the

---

[2]Revision 1 of Regulatory Guide 1.153, "Criteria for Safety Systems," endorses IEEE Std 603-1991, "Criteria for Safety Systems for Nuclear Power Generating Stations," as a method acceptable to the NRC staff for satisfying the NRC's regulations with respect to the design, reliability, qualification, and testability of the power, instrumentation, and control portions of the safety systems of nuclear power plants.

[3]IEEE publications may be obtained from the IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854.

[4]"The term "safety systems" is synonymous with "safety-related systems." The General Design Criteria cover systems, structures, and components "important to safety." The scope of this draft regulatory guide is, however, limited to "safety systems," which are a subset of "systems important to safety."

Office of Nuclear Reactor Regulation in the review of applications to construct and operate nuclear power plants. This regulatory guide will apply to Chapter 7 of that document.

Regulatory guides are issued to describe and make available to the public such information as methods acceptable to the NRC staff for implementing specific parts of the NRC's regulations, techniques used by the staff in evaluating specific problems or postulated accidents, and guidance to applicants. Regulatory guides are not substitutes for regulations, and compliance with regulatory guides is not required. Regulatory guides are issued in draft form for public comment to involve the public in the early stages of developing the regulatory positions. Draft regulatory guides have not received complete staff review and do not represent official NRC staff positions.

The information collections contained in this draft regulatory guide are covered by the requirements of 10 CFR Part 50, which were approved by the Office of Management and Budget, approval number 3150-0011. The NRC may not conduct or sponsor, and a person is not required to respond to, a collection of information unless it displays a currently valid OMB control number.

B. DISCUSSION

The use of industry consensus standards is part of an overall approach to meeting the requirements of 10 CFR Part 50 when developing safety systems for nuclear power plants. Compliance with standards does not guarantee that regulatory requirements will be met. However, compliance does ensure that practices accepted within various technical communities will be incorporated into the development and quality assurance processes used to design safety systems. These practices are based on past experience and represent industry consensus on approaches used for development of such systems.

Software incorporated into instrumentation and control systems covered by Appendix B will be referred to in this regulatory guide as safety system software. The software requirements specification is an essential part of the record of the design of safety system software. Associated with system requirements allocated to software subsystems, software requirements serve as the design bases for the software to be developed. Therefore, software requirements specifications are a crucial design input to the remainder of the software development process. Software requirements specifications should exhibit characteristics, such as correctness and completeness, that will facilitate the

implementation of a carefully planned and controlled software development process.

One consensus standard on software engineering, IEEE Std 830-1993, describes current practice for writing software requirements specifications for a wide variety of systems. It is not specifically aimed at safety applications; however, it does provide guidance on the development of software requirements specifications that will exhibit characteristics important for developing safety system software. This is consistent with the NRC staff's goals of ensuring high-integrity software in reactor safety systems.

Other standards mention software requirements specifications but do not provide detailed guidance for writing them. IEEE Std 7-4.3.2-1993, "Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations,"[3] which is endorsed by Revision 1 of Regulatory Guide 1.152, "Criteria for Digital Computers in Safety Systems of Nuclear Power Plants," mentions unambiguous software requirements as a prerequisite for high quality software development. IEEE Std 1012-1986, "IEEE Standard for Software Verification and Validation Plans,"[3] mentions unambiguous software requirements as a prerequisite for verification and validation. IEEE Std 1074-1991, "IEEE Standard for Developing Software Life Cycle Processes,"[3] describes software requirements specifications as an essential input at the beginning of a software development life cycle. Correct, complete, well-written and unambiguous software requirements are essential inputs to the main design and verification processes that are accepted as necessary to produce high-integrity software products [see NUREG/CR-6101, "Software Reliability and Safety in Nuclear Reactor Protection Systems" (November 1993),[5] and NUREG/CR-6263, "High Integrity Software for Nuclear Power Plants: Candidate Guidelines, Technical Basis and Research Needs" (June 1995)].

C. REGULATORY POSITION

The recommended practices in IEEE Std 830-1993 provide an approach acceptable to the NRC staff for meeting the requirements of 10 CFR Part 50 as

---

[5]Copies are available at current rates from the U.S. Government Printing Office, P.O. Box 37082, Washington, DC 20402-9328 (telephone (202)512-2249); or from the National Technical Information Service by writing NTIS at 5285 Port Royal Road, Springfield, VA 22161. Copies are available for inspection or copying for a fee from the NRC Public Document Room at 2120 L Street NW., Washington, DC; the PDR's mailing address is Mail Stop LL-6, Washington, DC 20555; telephone (202)634-3273; fax (202)634-3343.

they apply to the preparation of software requirements specifications for safety system software, subject to the provisions listed below.

To meet the requirements of 10 CFR 50.55a(h) and Appendix A to 10 CFR Part 50 as assured by complying with the criteria of Appendix B applied to the verification, validation, reviews, and audits of software used in or affecting basic components of nuclear power plants, the following provisions are necessary and will be considered by the NRC staff in the review of applicant submittals. (In this Regulatory Position, the cited criteria are in Appendix A or B of 10 CFR Part 50 unless otherwise noted.)

1. Section 3 of IEEE Std 830-1993 refers to IEEE Std 610.12-1990, "IEEE Standard Glossary of Software Engineering Terminology,"[3] for definitions of technical terms. These definitions are acceptable with the following clarifications or additions.

   1.1 Baseline

   Meaning 1 of baseline in IEEE Std 610.12-1990 is to be used in IEEE Std 830-1993. Formal review and agreement is taken to mean that responsible management has reviewed and approved a baseline.

   1.2 Interface

   All four variations of meaning in IEEE 610.12-1990 are to be used in IEEE Std 830-1993, depending on the context. Meaning 1, "A shared boundary across which information is passed," is interpreted broadly according to Criterion III to include design interfaces between participating design organizations.

2. Section 4.3 of IEEE Std 830-1993 defines a set of characteristics of a good software requirements specification (SRS). The first sentence of this section should be modified to read "An SRS must be...." The following clarifications and additional information should be provided for this set of characteristics for safety system software.

   2.1 Unambiguous

   When specification or representation tools are used for requirements, as described in sections 4.3.2.2 and 4.3.2.3 of IEEE Std 830-1993, traceability should be maintained between these representations and the natural language

descriptions of the software requirements that are derived from system requirements and system safety analyses.

2.2 Completeness

For safety system software, the description of functional requirements should include a specification of how functions are initiated and terminated as well as the system status at termination. Accuracy requirements, including units, error bounds, data type, and data size, should be provided for each input and output variable. Variables controlled or monitored in the physical environment should be fully described. Functions expressly prohibited should also be described.

Timing information is particularly important in specifying software requirements for safety system software. Functions with timing constraints should be identified and criteria for each mode of operation should be provided. Timing requirements should be deterministic and specified for both normal and anticipated failure conditions.

2.3 Consistent

IEEE Std 830-1993 restricts the term to mean internal consistency, noting that an external inconsistency is actually an incorrect specification of a requirement. The term is used in this regulatory guide to mean both internal and external consistency. External consistency implies that the SRS is consistent with associated software products and system products, such as safety system requirements and design. Internal consistency means that no requirement in the requirements specification conflicts with any other requirement in the specification.

2.4 Ranked for Importance or Stability

For safety system software, this characteristic means that software requirements important to safety be identified as such in the SRS. Criterion 20 of Appendix A, among others, describes the functions that reactor protection systems must perform. Section 4.3.5.2 of IEEE Std 830-1993 suggests three degrees of necessity of requirement: *essential*, *conditional*, and *optional*. As used in IEEE Std 830-1993, the terms conditional and optional refer to requirements that are not necessary for the software to be acceptable. For safety system software, unnecessary requirements should not be imposed. There may be documented variations in

essential requirements, but the variations must be linked in the software requirements specifications either to site and equipment variations or to specific plant design bases and regulatory provisions.

2.5 Verifiable

IEEE Std 830-1993 recommends the removal or revision of unverifiable requirements. This is clarified to mean that all requirements should be verifiable and should be modified or restated as necessary so that it is possible to verify each one.

2.6 Modifiable

This term is closely related to the style (form, structure, and modularity), readability, and understandability of the SRS. With respect to these characteristics, it is important that precise definitions of technical terms be available, either in the SRS or in a glossary.

2.7 Traceable

Section 4.3.8 of IEEE Std 830-1993 describes two types of traceability. Both types of traceability are required. Each identifiable requirement in an SRS must be traceable backwards to the system requirements and the design bases or regulatory requirements that it satisfies. Each identifiable requirement should be written so that it is also "forward traceable" to subsequent design outputs, e.g., from SRS to software design and from software design to SRS.

Forward traceability to all documents spawned by the SRS includes verification and validation materials. For example, a forward trace should exist from each requirement in the SRS to the specific inspections, analyses, or tests used to confirm that the requirement has been met.

3. Section 4.5(b) of IEEE Std 830-1993 recommends that SRSs be baselined and subject to a formal process for control of changes. The SRS must be subject to control of changes. Although this could be met directly by a change control procedure unique to IEEE Std 830-1993, it may also be accomplished by taking the SRS under a general software configuration management program as a configuration item.

4. Any entry in an SRS that is incomplete (uses "TBD"), as described by section 4.3.3.1 of IEEE Std 830-1993, must describe the applicable design bases and commitments to standards or regulations that govern the final determination of the requirement entry.

5. Section 4.7 of IEEE Std 830-1993 recommends that design-specific issues such as module partitioning, function allocation, and information flow be omitted from SRSs. Section 4.7.1 of IEEE Std 830-1993 states some exceptions to this policy, including reasons of security or safety. When specific design techniques or features such as independence, separation, diversity, and defense-in-depth are required by the safety system design bases or by regulation, these are an appropriate part of an SRS and they should be described therein.

6. Section 5.3.6 of IEEE Std 830-1993 lists software attributes that can serve as requirements. Attributes of particular interest for safety system software are safety, security, and reliability or robustness.

6.1 Safety

Software requirements important to safety are derived from system requirements and safety analyses and should be identified as such in the SRS. These requirements should include considerations based on the safety analysis report (SAR) as well as abnormal conditions and events (ACEs) as described in IEEE Std 7-4.3.2-1993, as endorsed by Revision 1 of Regulatory Guide 1.152.

6.2 Security

Security threats to the computer system should be identified and classified according to impact on safety and likelihood of occurrence. Actions required of the software to detect, prevent, or mitigate such security threats should be specified, including access control restrictions. For instance, modification of instrument calibration data might be protected by a password system.

6.3 Robustness

Requirements for fault tolerance and failure modes should be specified for each operating mode. Software behavior in the presence of unexpected, incorrect, anomalous, and improper input, hardware behavior, or software behavior should be fully specified. Software requirements for handling both hardware and software failures should be provided, including requirements for analysis of and recovery from computer system failures. Requirements for on-line in-service testing and diagnostics should be specified.

7. Because of its generality, IEEE Std 830-1993 discusses or recommends a number of content items that may be inappropriate to real-time, embedded safety systems. Headings for such inappropriate subjects in an SRS that is compliant with IEEE Std 830-1993 should be listed, followed by "Not applicable." For example, a graphical user interface may be inappropriate for a real-time, embedded reactor trip system.

8. Annex A to IEEE Std 830-1993 is not endorsed by this regulatory guide and may be taken only as examples. Directions to use an outline from Annex A, such as those directions found in section 5.3.7 of IEEE Std 830-1993, may be taken as advisory only.

9. Various sections in IEEE Std 830-1993 reference several industry codes and standards. These referenced standards should be considered individually. If a referenced standard has been incorporated separately into the NRC's regulations, licensees and applicants must comply with that standard as set forth in the regulation. If the referenced standard has been endorsed in a regulatory guide, the standard constitutes a method acceptable to the NRC staff of meeting a regulatory requirement as described in the regulatory guide. If a referenced standard has been neither incorporated into the NRC's regulations nor endorsed in a regulatory guide, licensees and applicants may consider and use the information in the referenced standard, if appropriately justified, consistent with current regulatory practice.

D. IMPLEMENTATION

The purpose of this section is to provide information to applicants and licensees regarding the NRC staff's plans for using this regulatory guide. No backfitting is intended or approved in connection with the issuance of this proposed guide. Any backfitting that may result from applying this new guidance to operating plants would be justified in accordance with established NRC backfitting guidance and procedures.

This draft guide has been released to encourage public participation in its development. Except in those cases in which an applicant proposes an acceptable alternative method for complying with specified portions of the NRC's regulations, the method to be described in the active guide reflecting public comments will be used in the evaluation of submittals in connection with applications for construction permits, standard design certifications and design approvals, and combined operating licenses. The active guide will also be used to evaluate submittals from operating reactor licensees that propose modifications that go beyond the current licensing basis if those modifications are voluntarily initiated by the licensee and there is a clear connection between the proposed modifications and this guidance. This guide will be used in conjunction with, and will eventually be reflected in, the Standard Review Plan, which is currently under revision.

REGULATORY ANALYSIS

1. PROBLEM

Because traditional and well-understood methods of design and quality assurance for developing and manufacturing hardware apply imperfectly to software design and development, additional guidance beyond standard approaches for hardware is necessary if the intent of the NRC's regulations is to be achieved. This problem is faced in many industries as computers and software replace traditional hardware-only instrumentation and control (I&C) designs. To this extent, the nuclear industry is not very different from any industry associated with high-consequence hazards. While additional guidance is necessary to help prevent failures of digital I&C safety systems, the potential benefits of these systems make their use highly desirable.

The use of computers and software in safety-related I&C designs is part of the larger problem of ensuring long-term safety of nuclear power plants, and it is seen as part of the solution as well. It is not just digital systems themselves that give rise to concerns about design verification and quality assurance, but the increase in complexity of the system designs (including software) being attempted is also a factor. The NRC staff discussed its concerns in SECY 91-292, "Digital Computer Systems for Advanced Light Water Reactors,"[1] and again in parts of SECY 93-087, "Policy, Technical, and Licensing Issues Pertaining to Evolutionary and Advanced Light-Water Reactor (ALWR) Designs."[1] Subsequently, the staff sponsored studies that resulted in characterization of design factors, guidelines, technical bases, and practices generally considered appropriate for high-integrity software [see NUREG/CR-6101, "Software Reliability and Safety in Nuclear Reactor Protection Systems" (November 1993); NUREG/CR-6113, "Class 1E Digital Systems Studies" (October 1993); NUREG/CR-6263, "High Integrity Software for Nuclear Power Plants: Candidate Guidelines, Technical Basis and Research Needs" (June 1995); NUREG/CR-6293, "Verification and Validation Guidelines for High Integrity Systems" (March 1995); and NUREG/CR-6294, "Design Factors for Safety-Critical Software" (December 1994)[2]]. These studies

---

[1]Copies are available for inspection or copying for a fee from the NRC Public Document Room at 2120 L Street NW., Washington, DC; the PDR's mailing address is Mail Stop LL-6, Washington, DC 20555; telephone (202)634-3273; fax (202)634-3343.

[2]Copies are available at current rates from the U.S. Government Printing Office, P.O. Box 37082, Washington, DC 20402-9328 (telephone (202)512-2249); or from the National Technical Information Service by writing NTIS at 5285 Port Royal Road, Springfield, VA 22161. Copies are available for inspection or copying for a fee from

identified software design control techniques that are currently being used in "best practice" software development efforts. While it is possible to simply list the criteria covered, the problem still remains of reaching a common understanding between NRC staff and industry practitioners regarding what constitutes acceptable software engineering practice for safety systems. An agreed-upon collection of standards, established practice, and engineering techniques for software engineering methods is needed to complement the collection that already supports traditional hardware engineering methods, such as statistical quality control, testing standards, and quality assurance techniques, used on design and manufacturing processes for hardware components.

Good software requirements specifications (SRS) have been identified by many studies as a key prerequisite for the production of high-integrity systems. Some authors estimate that fully 50% of software errors originate during the requirements phase of development, and the importance of good software requirements was noted in all of the references cited above. While controlling the format of an SRS cannot, by itself, improve the quality of the requirements described by the SRS, a disciplined process for producing the SRS probably will improve the quality. This is consistent with the requirement of Criterion III of Appendix B for discipline and control in the design process. Consequently, the importance of software requirements to high-integrity systems, in combination with the regulatory requirement for discipline and control in the design process, makes a disciplined process for producing software requirements specifications an appropriate subject for NRC staff review.

2. ALTERNATIVE APPROACHES

Based on the studies cited above, an alternative was identified in which consensus in the software engineering community is sufficient to ensure widespread familiarity and reasonable levels of agreement. There are two additional approaches, taking no action and prescribing a detailed approach built from staff selections of best practice. In all, three approaches were identified.

---

the NRC Public Document Room at 2120 L Street NW., Washington, DC; the PDR's mailing address is Mail Stop LL-6, Washington, DC 20555; telephone (202)634-3273; fax (202)634-3343.

1. Take no action,
2. Prescribe a detailed approach,
3. Endorse one or more software engineering standards.

The first alternative, taking no action, has the advantage that its initial cost is low since there are no "start-up" activities. It has flexibility, since each applicant would develop its own technical basis demonstrating that its digital system, and the quality assurance measures applied to it, complies with the NRC's regulations. However, this could have adverse effects on the level of staff effort required to conduct reviews or to ensure consistency among reviewers. In the absence of an identified set of commonly accepted guidelines, practices, and quality assurance measures applicable to software engineering, NRC staff review would take longer and require greater effort to ensure consistent staff safety evaluations. From the applicant's perspective, this flexibility also has associated potential costs because there could be more unknowns associated with demonstrating compliance with regulations. Although the initial cost would apparently be low, taking no action could result in greater total costs, to both the NRC staff and the applicant, during the safety evaluation process.

Prescribing a detailed approach could have significant preliminary costs involved in formulating the approach and dealing with the public comment that would inevitably result. The staff has been reluctant in the past to take this approach. Such an approach places the staff in the position of designer, and compromises, or appears to compromise, the staff's independence as design safety reviewers; this is not the role of the regulator.

Consensus standards on software development are available, and they represent current good practice as agreed upon by responsible professionals in the software industry. Many organizations issuing standards, such as the IEEE and ANSI, provide for review and revision of standards at regular intervals to ensure the consensus positions are current. In the United States, the Institute of Electrical and Electronic Engineers (IEEE), the American Nuclear Society (ANS), the Electronic Industry Association (EIA), the Instrument Society of America (ISA), the American Society of Mechanical Engineers (ASME), and the American National Standards Institute (ANSI) are the standards bodies issuing software engineering standards, computer standards, or related quality standards. In Europe, the International Electrotechnical Commission (IEC), the International Organization for Standardization (ISO), the International Atomic Energy Agency

(IAEA), and the Comité Consultatif International Télégraphique et Téléphonique (CCITT) fill the same roles. The European Committee for Electrotechnical Standardization (CENELEC), a regional standardization body, adopts national and international standards. The overall collection of standards issued by these bodies covers a variety of subjects considered important to software quality.

The standards specific to the nuclear industry issued by the U.S.-based organizations are, in general, compatible with the NRC's regulations. The software engineering standards issued by these organizations, notably the IEEE software engineering standards, are in general compatible with nuclear-industry-specific standards. Together, these standards form a framework for addressing the use of software within nuclear systems in the U.S. nuclear regulatory environment. Selected international standards can complement this framework; however, they tend to be organized differently and do not map directly into the U.S. industry-specific framework.

3. VALUES AND IMPACTS

Values and impacts for each of the three identified approaches are analyzed below. In this analysis, the probability of an alternative approach having a positive effect on software quality and the probability of the effect of software quality on the achievement of overall safety goals are not known quantitatively. Although the current state of the art does not support quantitative estimates, the results of poor software quality are evident in notable instances of software failure in various industries. Therefore, a positive correlation between software quality and the achievement of safety goals is inferred from the instances of negative effects of poor software quality, i.e., software quality is a necessary but insufficient factor in achieving safety goals. In the summary below, an impact is a cost in schedule, budget, or staffing or it is an undesired property or attribute that would accrue from taking the proposed approach. Both values and impacts may be functions of time.

3.1 Alternative 1 - Take No Action

If no action is taken, retaining the status quo, the NRC staff will continue to receive applications or requests to review safety questions that have been prepared with no clear guidance on what the staff considers to be acceptable methods of ensuring that safety-related software meets the requirements of the NRC's regulations. Each applicant would propose such measures as it deems

necessary, which would be reviewed by the staff and discussed with the applicant to reach a resolution ultimately acceptable to the staff and the NRC. This preserves the value of flexibility, but at the impact of additional effort on the part of the NRC staff and applicants, as well as potential schedule extensions. It is possible that a de facto staff position would develop from the accumulation of successful applications, but the amount of time and effort required to reach this condition is unknown.

Value - No value beyond the status quo

Impact - Schedule, budget, and staffing cost, to the staff and applicant, associated with regulatory uncertainty

3.2 Alternative 2 - Prescribe a Detailed Approach

If the staff prescribed a detailed approach, applicants would enjoy regulatory certainty at the expense of reduced flexibility. Immediate tangible impacts would include staff time to specify and defend the approach in a public forum. Intangible impacts would include a potential compromise of staff effectiveness as impartial safety reviewers and a loss of input from innovative applicants. Future impacts would include continual review of the approach as newer software engineering methods are developed by the technical community.

Values - Probable improvement in the likelihood of achieving safety goals as a consequence of staff expertise and specialized knowledge derived during the development of the prescribed approach

- Common understanding of regulatory view of software practice

Impacts - Cost of staff effort to develop the approach

- Potential compromise of staff objectivity
- Innovative approaches discouraged as a result of increased cost
- Cost of evolving, maintaining, and communicating the approach

3.3 Alternative 3 - Endorse One or More Software Engineering Standards

If the staff endorses selected consensus software engineering standards, the staff and applicants obtain the benefit of the work of responsible software engineering professional standards committee volunteers. The value in this is the common understanding between the staff and applicants of an approach that has acceptance as good practice in the technical community. The standards usually permit tailoring to meet the needs of particular situations, so a medium level of flexibility is retained. Additional staff effort is minimal, since members of the staff are already active in standards committees that the staff considers

important to safety. Because detailed standards that address specific software engineering practices are available, the staff may select standards that address topics of particular importance regarding safety system software. Many standards, including IEEE software engineering standards, are reviewed and updated periodically, which acquaints the staff with changing practices. Coordination of standards efforts for standards used widely in the United States with international standards efforts is increasing, but the outcome of this is still unpredictable.

Values
- Probable improvement in the likelihood of achieving safety goals as a consequence of improvement in software practices
- Consideration of relevant topics
- Common understanding of good software practice, as defined by consensus processes in the software industry
- Maintenance and evolution of the definition of good software practice by the software industry

Impact
- Cost of endorsing the selected standards

4. CONCLUSIONS

There are a number of potential benefits associated with the use of digital I&C safety systems in nuclear power plants. Implementation of these systems must be consistent with the NRC's regulations. Three approaches to providing additional guidance for software were examined. Taking no action may result in accumulating regulatory expense as applicants submit proposed methods to assure the staff that safety-related software meets the requirements of the NRC's regulations. A de facto acceptable method would probably evolve, but the time and effort required for this to happen are unknown. A detailed staff prescription has unacceptable impacts and would involve the staff directly in the applicant's solution of technical problems. Endorsing selected software engineering standards has good value with minimal impact and addresses the stated problem. Note that none of these approaches presents new regulatory requirements; they define acceptable approaches for meeting existing requirements.

5. DECISION RATIONALE

Based on the lowest impact and highest value for problem solution capability, the third alternative, endorsing selected software engineering standards, has been chosen. The highest value will be achieved by selecting standards that address software engineering processes that have a high potential for ensuring that safety system software meets the requirements of the NRC's regulations as applied to software. Standards should be selected based upon relevance and maturity.

BIBLIOGRAPHY

Hecht, H., A.T. Tai, K.S. Tso, "Class 1E Digital Systems Studies, NUREG/CR-6113, USNRC, October 1993.[1]

Hecht, H., et al., "Verification and Validation Guidelines for High Integrity Systems," NUREG/CR-6293, USNRC, March 1995.[1]

Institute of Electrical and Electronics Engineers, "Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations," IEEE Std 7-4.3.2, 1993.

Lawrence, J.D., "Software Reliability and Safety in Nuclear Reactor Protection Systems," NUREG/CR-6101 (UCRL-ID-117524, Lawrence Livermore National Laboratory), USNRC, November 1993.[1]

Lawrence, J.D, and G.G. Preckshot, "Design Factors for Safety-Critical Software," NUREG/CR-6294, USNRC, December 1994.[1]

Seth, S., et al., "High Integrity Software for Nuclear Power Plants: Candidate Guidelines, Technical Basis and Research Needs," NUREG/CR-6263, USNRC, June 1995.[1]

USNRC, "Criteria for Digital Computers in Safety Systems of Nuclear Power Plants, Regulatory Guide 1.152, Revision 1, January 1996.[2]

USNRC, "Standard Review Plan," NUREG-0800, February 1984.

---

[1]Copies may be purchased at current rates from the U.S. Government Printing Office, P.O. Box 37082, Washington, DC 20402-9328 (telephone (202)512-2249); or from the National Technical Information Service by writing NTIS at 5285 Port Royal Road, Springfield, VA 22161. Copies are available for inspection or copying for a fee from the NRC Public Document Room at 2120 L Street NW., Washington, DC; the PDR's mailing address is Mail Stop LL-6, Washington, DC 20555; telephone (202)634-3273; fax (202)634-3343.

[2]Single copies of regulatory guides may be obtained free of charge by writing the Office of Administration, Attention: Distribution and Services Section, U.S. Nuclear Regulatory Commission, Washington, DC 20555-0001; or by fax at (301)415-2260. Copies are available for inspection or copying for a fee from the NRC Public Document Room at 2120 L Street NW, Washington, DC; the PDR's mailing address is Mail Stop LL-6, Washington, DC 20555; telephone (202)634-3273; fax (202)634-3343.

Federal Recycling Program

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, DC 20555-0001

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

FIRST CLASS MAIL
POSTAGE AND FEES PAID
USNRC
PERMIT NO. G-67