Issues and Relationships Among Software Standards for Nuclear Safety Applications
Version 2.0

J. A. Scott
G. G. Preckshot
J. D. Lawrence
G. L. Johnson

March 26, 1996

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the Laboratory.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED
DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This report has been reproduced
directly from the best available copy.

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information
P.O. Box 62, Oak Ridge, TN 37831
Prices available from (615) 576-8401, FTS 626-8401

Available to the public from the
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.,
Springfield, VA 22161

This work was supported by the United States Nuclear Regulatory Commission under a Memorandum of Understanding with the United States Department of Energy, and performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.
Issues and Relationships Among Software Standards for Nuclear Safety Applications

Version 2.0

Prepared by
J. A. Scott
G. G. Preckshot
J. D. Lawrence
G. L. Johnson

Prepared for
U.S. Nuclear Regulatory Commission

FESSP
Fission Energy and Systems Safety Program
Lawrence Livermore National Laboratory
ABSTRACT

Lawrence Livermore National Laboratory is assisting the Nuclear Regulatory Commission with the development of draft regulatory guides for selected software engineering standards. This report describes the results of the initial task in this work. The selected software standards and a set of related software engineering standards were reviewed, and the resulting preliminary elements of the regulatory positions are identified in this report. The importance of a thorough understanding of the relationships among standards useful for developing safety-related software is emphasized. The relationship of this work to the update of the Standard Review Plan is also discussed.
## CONTENTS

**PREFACE** .................................................................................................................................................. vii

**EXECUTIVE SUMMARY** ........................................................................................................................... ix

**1.0 INTRODUCTION** .................................................................................................................................... 1
  1.1 Purpose of the Standards Review Task .................................................................................................. 1
  1.2 Approach .............................................................................................................................................. 2
  1.3 Acronyms .............................................................................................................................................. 3
  1.4 Contents of this Report ......................................................................................................................... 4

**2.0 RELATIONSHIPS AMONG STANDARDS** ............................................................................................ 4
  2.1 Frameworks of Standards .................................................................................................................... 4
  2.2 Dimensions of Standards Relationships ............................................................................................... 5
    2.2.1 Authority ...................................................................................................................................... 5
    2.2.2 Criticality ................................................................................................................................... 5
    2.2.3 Industry Focus ............................................................................................................................... 5
    2.2.4 Life Cycle ................................................................................................................................... 5
    2.2.5 Special Topics ............................................................................................................................... 5
    2.2.6 Scope Tailoring ............................................................................................................................. 5
    2.2.7 Software Technology Orientation ............................................................................................... 6
    2.2.8 System Level ................................................................................................................................ 6
  2.3 Standards Frameworks .......................................................................................................................... 6
    2.3.1 Key Standards in Use in the U.S. ..................................................................................................... 6
    2.3.2 International Standards Activities ............................................................................................... 10
  2.4 Framework Issues .................................................................................................................................. 11
    2.4.1 U.S. Framework ............................................................................................................................ 11
    2.4.2 International Framework ............................................................................................................... 12

**3.0 DISCUSSION OF THE STANDARDS REVIEW TASK ANALYSES** ......................................................... 14
  3.1 Software Configuration Management (ANSI/IEEE Std 828 and ANSI/IEEE Std 1042) ....................... 14
    3.1.1 Standard Status and Future Directions .......................................................................................... 14
    3.1.2 Conflicts and Approaches ............................................................................................................. 15
  3.2 Software Verification and Validation (ANSI/IEEE Std 1012 and IEEE Std 1028) ............................... 16
    3.2.1 Standard Status and Future Directions .......................................................................................... 16
    3.2.2 Conflicts and Approaches ............................................................................................................. 16
  3.3 Software Requirements (ANSI/IEEE Std 830) .................................................................................... 18
    3.3.1 Standard Status and Future Directions .......................................................................................... 18
    3.3.2 Conflicts and Approaches ............................................................................................................. 18
  3.4 Software Test Documentation (ANSI/IEEE Std 829) ........................................................................... 19
    3.4.1 Standard Status and Future Directions .......................................................................................... 19
    3.4.2 Conflicts and Approaches ............................................................................................................. 19
  3.5 Software Unit Testing (ANSI/IEEE Std 1008) ....................................................................................... 19
    3.5.1 Standard Status and Future Directions .......................................................................................... 19
    3.5.2 Conflicts and Approaches ............................................................................................................. 20
PREFACE

This report is an updated version of a letter report provided to the Nuclear Regulatory Commission in 1995. The original intent was to summarize activities and future plans for Lawrence Livermore National Laboratory's support of the NRC development of regulatory guides for software engineering standards. Since much of the material is of more general interest, this revision has been prepared for publication. Some of the material, such as the description of working group activities, has not been updated since it is not important to the intent of this version.
EXECUTIVE SUMMARY

Lawrence Livermore National Laboratory (LLNL) is assisting the Nuclear Regulatory Commission (NRC), Office of Nuclear Regulatory Research, with the development of regulatory guides endorsing selected software engineering standards. This work is related to another LLNL effort supporting the NRC in an update of Chapter 7 of the Standard Review Plan (SRP). The work to support the SRP update is being performed for the NRC Office of Nuclear Reactor Regulation. The software engineering technical basis supporting the SRP update is reflected, in part, in a set of software engineering standards. The regulatory guides to be produced will endorse the key standards and communicate methods acceptable to the NRC staff for meeting the Commission's regulations with respect to software in digital I&C safety systems. This report describes the results of the first task, which provided a high-level examination of the consistency of the key standards and identified the issues to be addressed in the regulatory guides.

Safety-related software developed for nuclear power plants must meet regulatory requirements for structures, systems, and components, including 10 CFR Part 50 Appendix A, "General Design Criteria for Nuclear Power Plants" and 10 CFR Part 50 Appendix B, "Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants." To meet these requirements, software must perform with high integrity. Adherence to standards in the development of systems and software does not guarantee high integrity; however, it does ensure that currently accepted practices will be used. In general, standards represent industry consensus on the state of the practice and provide an accepted basis for structuring software development processes. Structuring an effective software development process based on standards requires familiarity with the requirements of individual standards as well as the relationships among the standards. A thorough understanding of these considerations is referred to in this report as "understanding the standards framework." In general, the standards in the "U.S. framework," although perhaps not planned as a framework, fit together reasonably well. Industry standards, such as ASME NQA-1 and IEEE standards sponsored by the Nuclear Power Engineering Committee of the IEEE Power Engineering Society (e.g., IEEE Std 603 and IEEE Std 7-4.3.2), focus on system aspects, while the software standards can be used to add to or elaborate on the requirements of these standards.

Several important results have emerged from this work. The importance, to both developers and reviewers, of a thorough understanding of the standards framework associated with software development has been underscored. Standards within a framework relate to each other in a number of dimensions, all of which must be understood in order to make effective use of the standards. In addition to understanding the framework in use in the U.S., there are a variety of reasons for monitoring and participating in the development of international standards. These include preventing serious potential conflicts between frameworks, understanding products developed in other frameworks, and preparing for the possible future adoption of international standards.

The original list of candidate standards included IEC 880-1986. Significant differences from the U.S. standards framework were noted in the areas of scope and prescriptiveness. As a result, this standard was removed as a standard to be endorsed in the initial set of regulatory guides and replaced with ANSI/IEEE Std 1074-1991. The resulting set of software engineering standards is generally compatible with the nuclear industry standards in use in the U.S.
1.0 INTRODUCTION

Lawrence Livermore National Laboratory (LLNL) is currently performing research for the Nuclear Regulatory Commission (NRC), Office of Nuclear Regulatory Research, on an effort entitled “Software Regulatory Guide Development.” The objective of this research is to provide guidance regarding high-integrity software by assisting the NRC in developing regulatory guides endorsing selected software engineering standards. This research is related to another LLNL effort supporting the NRC in an update of Chapter 7 of the Standard Review Plan (SRP). The work to support the SRP update is being performed for the NRC Office of Nuclear Reactor Regulation. Part of the SRP update effort involves providing guidance on software engineering process considerations for the design and development of digital instrumentation and control (I&C) systems in nuclear power-generating stations. The software engineering technical basis supporting the SRP update is reflected, in part, in a set of software engineering standards. In the Software Regulatory Guide Development project, regulatory guides will be drafted to communicate methods acceptable to the NRC staff for meeting the Commission’s regulations as applied to software in digital I&C safety systems. This report describes the results of the first task, the Standards Review task, performed on this project.

1.1 Purpose of the Standards Review Task

The purpose of the Standards Review task was to provide an overview of the relationships and potential conflicts among eight software engineering standards identified for NRC endorsement by regulatory guides. An additional goal was to examine the relationships of the eight selected standards to a set of 15 related standards. This examination of relationships was intended to identify major areas where exceptions, additions, or clarifications would be necessary in the regulatory guides, and to identify the probable approaches to be taken in the development of the specific regulatory guides. The goal was to ensure that any existing major conflicts would be discovered and addressed before developing drafts of specific regulatory guides.

Regulatory guides were initially planned for the following standards. The analyses performed in the Standards Review task indicated that IEC 880 differed significantly from the other standards in scope and level of prescriptiveness and that ANSI/IEEE Std 1074 provides an important “organizing” role for the software engineering standards. Consequently, IEC 880 was dropped as a candidate standard and ANSI/IEEE Std 1074 was added.

- ANSI/IEEE Std 830-1993: IEEE Recommended Practice for Software Requirements Specifications
The related standards are:

- ANSI/ANS 10.4-1987
- IEEE Std 730.1-1989
- IEEE Std 982.1-1988
- IEEE Std 1016-1987
- IEEE Std 1044-1993
- IEEE Std 1061-1992
- IEEE Std C37.1-1987
- IEEE Std 1298-1992 Part I

An additional element of this task was an analysis of available preliminary drafts of material from working groups on standards that are undergoing revision. (The working groups referenced in this report may or may not be currently active.) The goal of this task element was to assess probable future directions of the selected standards and to evaluate the effects these might have on the proposed regulatory guides. Preliminary information, which has not yet been successfully balloted by the standards committees, would not be included directly in a regulatory guide. However, current trends could indicate matters on which the proposed regulatory guides should remain silent or should include flexibility.

1.2 Approach

The analyses performed on the Standards Review task were based on a functional breakdown of the software engineering areas addressed by the standards identified for endorsement by regulatory guides. This breakdown is shown in Table 1.

The eight selected standards and the 15 related standards were examined with respect to each functional area. For each area, a set of keyword classifications was recorded and then cross-referenced to the standard passages bearing upon the keyword subject. Each such keyword entry could also include reviewer comments. This information was sorted by keyword, which placed relevant standard citations with reviewer comments in juxtaposition. This facilitated direct comparison of related clauses or sentences in each standard, and subsequent summarization. Notes on conflicts and probable approaches for dealing with the conflicts were included during the summary process. Finally, the summary information was cross-checked against the SRP update materials (including NUREG/CR references such as Lawrence, 1993) and a MITRE Corporation report addressing the technical basis and research needs associated with high-integrity software for nuclear power plants [Seth et al., 1995].
Table 1. Software Engineering Areas & Standards for Endorsement

<table>
<thead>
<tr>
<th>Software Activity or Topic</th>
<th>Type of Activity</th>
<th>Standards for Regulatory Guide Drafts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Requirements</td>
<td>Design</td>
<td>ANSI/IEEE Std 830-1993</td>
</tr>
<tr>
<td>Software Reviews &amp; Audits</td>
<td>Assurance</td>
<td>IEEE Std 1028-1988</td>
</tr>
<tr>
<td>Software in Safety Systems of Nuclear Power Plants (removed from consideration after the analyses)</td>
<td>Design</td>
<td>IEC 880-1986</td>
</tr>
</tbody>
</table>

1.3 Acronyms

ALWR  Advanced Light Water Reactor
ANS  American Nuclear Society
ANSI  American National Standards Institute
ASME  American Society of Mechanical Engineers
BTP  Branch Technical Position
CFR  Code of Federal Regulations
COTS  Commercial Off the Shelf
DOD  Department of Defense
I&C  Instrumentation and Control
IAEA  International Atomic Energy Agency
IEC  International Electrotechnical Commission
IEEE  Institute of Electrical and Electronic Engineers
ISO  International Organization for Standardization
LWR  Light Water Reactor
NRC  Nuclear Regulatory Commission
PESP  Pre-Existing Software Product
RBMK  Reactor Name (Russian abbreviation)
SRP  Standard Review Plan
1.4 Contents of this Report

Section 2 of this report presents a discussion of the general relationships among the standards examined. Section 3 presents the results of the analyses of each of the standards planned for endorsement by regulatory guides. Each subsection discusses conflicts and planned approaches as well as providing notes regarding working group activities. Section 4 presents conclusions based on these analyses. Appendix A provides a summary in tabular form of the relationships among the standards.

2.0 RELATIONSHIPS AMONG STANDARDS

2.1 Frameworks of Standards

Safety-related software developed for nuclear power plants must meet regulatory requirements for structures, systems, and components, including 10 CFR Part 50 Appendix A, “General Design Criteria for Nuclear Power Plants” and 10 CFR Part 50 Appendix B, “Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants.” The criteria of these Appendices apply to systems and related quality assurance processes and, if those systems include software, then the requirements extend to the software elements. To meet these requirements, as applied to software, safety-related software must perform with high integrity. However, demonstrating the achievement of high integrity levels in software developments can be difficult. Current software engineering practices do not guarantee the absence of faults in software. Therefore, for high-integrity nuclear software, “best software engineering practices” are applied to the development of software, and additional measures, such as diversity and defense-in-depth, are applied at the system level.

Adhering to standards in the development of systems and software does not guarantee high integrity; however, it does ensure that currently accepted practices will be used. In general, standards represent industry consensus on the state of the practice and provide an accepted basis for structuring software development processes. To structure an effective software development process based on standards, it is necessary to be familiar with the detailed requirements of individual standards as well as the relationships among the standards. In the former area, one needs to be concerned with such issues as the applicability and completeness of a given standard regarding the software to be developed while, in the latter area, one should consider consistency among the standards chosen and the extent to which the selected set of standards addresses software development needs. Finally, one should also be aware of any context assumptions implicit in the standard. An example is the contract orientation of the ISO 9000 standards, which is focused on satisfying agreements between customer and supplier and, hence, may not be appropriate where regulatory requirements apply.

In addition to the standards themselves, additional guidance should be sought in the regulatory positions taken with respect to specific, key standards. These regulatory positions could be reflected in NRC Regulatory Guides, the Standard Review Plan (NUREG-0800) and its associated Branch Technical Positions, or in NRC Generic Letters. Information provided in these regulatory positions pertains to specific questions regarding nuclear applications and, in some cases, modifies the applicability of a standard by adding to, taking exception to, or clarifying the meaning of some provisions of a standard.

A thorough understanding of these considerations is referred to in this report as “understanding the standards framework.” Multiple standards frameworks exist and variations of a given standards framework are likely in different software development environments. The important commonalities are the structuring of a standards-based, high-integrity software development process and a thorough understanding of the role played by the suite of related standards, and associated regulatory guidance, in the quality assurance process. A poor understanding of the standards framework and its relationship to a
proposed software development process may leave assurance objectives unfulfilled or expend effort without commensurate results.

2.2 Dimensions of Standards Relationships

Standards and their relationships can be analyzed along multiple dimensions. The dimensions described in this section were considered in the Standards Review task. A summary of the relationships among the examined standards, keyed to the dimensions discussed below, is presented in Appendix A. This appendix provides a brief overview or "road map" of the relationships among the standards examined.

2.2.1 Authority

Guidance contained in a standard can represent varying levels of authority. One way of categorizing standards by authority level is by Standard, Recommended Practice, and Guide. In situations where two or more standards discuss the same or a similar subject, some standards may cite or defer to others, giving a coordinated authority. The Standard level indicates that an official consensus within the standards-issuing body has been reached, and compliance requires that all of the standard's strictures ("shall") be observed. A Recommended Practice is similar to a Standard except that recommendations ("should") are described. Guides typically provide guidance on implementing a Standard ("how-to's").

2.2.2 Criticality

Some standards make reference to "critical" software and make special provisions for such software. The definitions of criticality vary from standard to standard. Industry-specific standards have more concrete definitions related to regulations and regulatory positions.

2.2.3 Industry Focus

Standards can be written for general application or for use in specific industries. Industry standards are specific to industry issues but can also vary in applicability, e.g., a focus on nuclear industry versus a focus on nuclear power plant safety systems.

2.2.4 Life Cycle

Some standards address activities that vary significantly over part or all of the product life cycle. Others, such as configuration management, are "integral activities" that vary only slightly with life cycle phase. Most standards that discuss a software life cycle do not require a specific life cycle model as long as the life cycle model in use can be mapped to the requirements of the standard.

2.2.5 Special Topics

A variety of special topics can be addressed by the standards comprising a given standards framework. Examples important to the NRC are the treatment of pre-existing software products (PESP) and system-related issues. Commercial off-the-shelf (COTS) software is one type of PESP.

2.2.6 Scope Tailoring

Standards with the authority level of Standard occasionally allow some tailoring in order to adjust the use of the standard to a particular situation. The term "tailoring" refers to the implementation of variations from, or additions to, the requirements of the standard that are permissible according to the standard, while still allowing the implementing organization to claim compliance with the standard. The permissible tailoring, if any, is defined in each standard.
2.2.7 Software Technology Orientation

Standards can be categorized by their software technology orientation, such as process, product, measurement, or reference (such as dictionaries).

2.2.8 System Level

Standards also vary by the portions of the overall system development process that they address. System-level standards address the requirements for the system and its components. Some standards address the relationship between system and software requirements, while others focus strictly on software according to a given software technology orientation.

2.3 Standards Frameworks

2.3.1 Key Standards in Use in the U.S.

A standards framework that is important for nuclear power plants in the U.S. includes ASME NQA-1-1994, “Quality Assurance Requirements for Nuclear Facility Applications,” and IEEE nuclear industry standards such as IEEE Std 603-1991, “Criteria for Safety Systems for Nuclear Power Generating Stations,” and IEEE Std 7-4.3.2-1993, “Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations.” In addition, various IEEE software engineering standards provide detailed guidance on software development. Figure 1 shows the relationships among 10 CFR Part 50 Appendices A and B, NRC Regulatory Guides, and the system-level standards in this framework. Figure 2 shows the interrelationships among the current versions of the standards referenced in Figure 1. It should be noted that NQA-1-1983 was endorsed by NRC Regulatory Guide 1.28 at a time when the NQA standards (NQA-1, NQA-2, and NQA-3) existed as separate entities. In the 1994 revision, these separate entities were combined and the aggregate was renamed NQA-1.

Figure 3 illustrates the roles played by nuclear system standards and software engineering standards in the framework for nuclear safety system software development. Regulatory requirements apply to systems, structures, and components, and provide the regulatory basis for the system-level standards in the framework. These standards are depicted in Figure 3 as breadth-oriented standards since they are concerned with compliance with the full range of the Commission’s regulations for safety systems. Since they are breadth-oriented, they do not typically address specific software issues in depth. Software engineering standards complement the breadth-oriented standards by providing industry consensus on the implementation of specific aspects of software engineering practice. Software engineering standards typically treat subjects in detail and, therefore, these standards are depicted in Figure 3 as depth-oriented standards. Compliance with a particular depth-oriented standard, within the context provided by nuclear regulation and system-level standards, will address a subset of the characteristics or attributes necessary for meeting regulatory requirements.

Figure 4 illustrates the roles played by specific standards in the framework. The block labeled System depicts IEEE Std 603 and IEEE Std 7-4.3.2 in their roles in support of system development. The former addresses computer and non-computer hardware elements while the latter addresses system-level issues for software. The block in Figure 4 that addresses design outputs shows the various software design activities and the specific standards in the framework that support those activities. ANSI/IEEE Std 1074 addresses the development of software life cycle processes and, therefore, serves to unify the individual activities standards. It also addresses assurance activities, referred to by ANSI/IEEE Std 1074 as integral processes. These are shown on the bottom of Figure 4. Finally, since NQA-1 addresses quality assurance for systems, based on 10 CFR 50 Appendix B, it applies to all activities shown in Figure 4.
Figure 1. Key Standards and Regulatory Guide Endorsement

Figure 2. Relationships Between Current Versions of the Key Standards
Figure 3. Elements of a Standards Framework for Nuclear Safety System Software Development
2.3.2 International Standards Activities

In addition to the standards framework described above, a number of international standards are of interest to the nuclear industry. Key standards in this framework are IEC 880, "Software for Computers in the Safety Systems of Nuclear Power Stations," IEC 987, "Programmed Digital Computers Important to Safety for Nuclear Power Stations," and IEC 1226, "Nuclear Power Plants — Instrumentation and Control Systems Important for Safety — Classification." These standards, as well as a standard currently in development to address the system-level aspects of these standards, originate from IEC Subcommittee SC45A (Reactor Instrumentation) of Technical Committee 45 (Nuclear Instrumentation). IEC 880 and IEC 987 reference the IAEA (International Atomic Energy Agency) Codes (safety standards) and Safety Guides. IEC 880 is related to the IAEA safety standards as shown in Figure 5. Technical Committee 65 (Industrial-Process Measurement and Control) has a subcommittee (SC65A: System Aspects) that is currently developing a standard entitled "Functional Safety: Safety-Related Subsystems" (IEC 1508). Because of the related interests of SC45A and SC65A, it is important that standards being developed and issued by SC65A be considered along with the nuclear standards of SC45A.

Another international standards area of interest is the work of the ISO/IEC Joint Technical Committee 1 (JTC1). JTC1 Subcommittee 7 deals with software development and system documentation. Different working groups of this subcommittee are addressing software issues including software quality (ISO/IEC 9126, "Information Technology — Software Product Evaluation — Quality Characteristics and Guidelines for Their Use"), software life cycle processes (ISO/IEC 12207, "Information Technology — Software Life Cycle Processes"), and the project on Software Process Improvement Capability Determination (SPICE). These activities will relate to software development in the international communities in ways similar to the IEEE software engineering standards in the framework described above for safety system software development in the U.S. SPICE addresses software capability assessment and is similar in scope to the Software Engineering Institute's Capability Maturity Model, but is structured somewhat differently.

Figure 5. IAEA Context and IEC 880
Finally, the quality standards of the ISO 9000 series, in particular ISO 9001 and the ISO 9000-3 guide for implementing ISO 9001 for software, must also be considered. ISO 9000 is contractually oriented and may have shortcomings regarding its use for developing high-integrity software; however, it is coming into widespread use and products developed to this standard might be proposed for future nuclear industry use. In any case, the ISO 9000 series would need to be used with the IAEA safety standards and safety guides or some other source of nuclear system design criteria.

2.4 Framework Issues

2.4.1 U.S. Framework

In general, the standards in use in the U.S. fit together reasonably well. Industry standards tend to be more focused on system aspects than the software standards and are more specific to nuclear issues. The software standards in this framework complement the requirements of the industry standards.

Several items should be noted regarding this standards framework. The first is that, although certain IEEE software engineering standards have been selected for endorsement by regulatory guides, standards issued by other bodies addressing various software engineering topics are also in use. An example is DOD-STD-2167A on Defense System Software Development. Consequently, it is possible that some software development organizations producing software within the regulatory purview of the NRC will have mature software development processes based on these standards. This is not a problem with respect to NRC regulatory guides, since they represent methods acceptable to the NRC staff but do not preclude other approaches, provided that they are consistent with the NRC regulations.

A second item of note is that standards frequently establish context and handle related topics by making reference to other standards. The referenced standards are useful for these purposes, however, a referenced standard is not necessarily acceptable to the NRC staff with respect to NRC regulations. Consequently, regulatory guides for software engineering standards will, as necessary, include specific instructions to the effect that endorsement of a given standard does not extend to referenced standards. Specific regulatory guidance will be issued for all standards that the NRC staff wishes to designate as representing acceptable methods for accomplishing various software engineering activities.

Finally, several items should be noted about specific standards. First, IEEE Std 7-4.3.2-1993 provides a link between systems and software issues and is, therefore, a fundamental element in the standards framework. This link is crucial for nuclear safety-related software since software safety requirements are derived from system safety requirements. An additional item of note concerns Subpart 2.7 of NQA-1-1994. The old ASME NQA-2a-1990, including Subpart 2.7, has not been endorsed by the NRC. However, it is referenced by IEEE Std 7-4.3.2-1993, which is endorsed by NRC Regulatory Guide 1.152, Revision 1. This reference might lead to questions regarding potential conflicts between Subpart 2.7 and the IEEE software engineering standards. Subpart 2.7 addresses activities in the software life cycle and, in particular, specifies various software verification and validation (V&V) activities. The IEEE standards tend to provide more extensive information and in some cases, such as software V&V, prescribe more activities than does Subpart 2.7. In considering the union of guidance on software development provided by all of the standards, this difference in activities specified by the various standards does not create inconsistencies with respect to the overall fabric of currently accepted software development practices. However, Subpart 2.7 uses the wording: "... phase software verification and validation activities shall consist of ..." in several places, which could be interpreted to mean that no other V&V activities are required. In this sense, the addition of requirements from the IEEE standards could be viewed as a conflict. However, since the IEEE software engineering standards reflect consensus about current

---

1A summary discussion of the relationship of software requirements to safety is given in Seth et al., 1995.
software practices, a product produced in conformance with only Subpart 2.7 will not necessarily have the benefit of current accepted practices, and must be considered carefully to determine what safety impact this lack may have.

2.4.2 International Framework

NRC endorsement of international standards for nuclear power plant computer systems should be of interest to the U.S. community for several reasons:

- There is a substantial body of international experience with class 1E digital systems that is applicable to U.S. systems.
- Plants certified to international standards in the U.S. will be more readily marketable overseas.
- Acceptance of equipment built to international standards for U.S. applications makes the products of international suppliers easier to apply to U.S. plants.
- International nuclear standards in the I&C area cover a broader range of topics than do U.S. nuclear standards.

Any standard endorsed by the NRC must, however, be adapted to be consistent with the extant regulatory and standards framework. This is a relatively simple task for the set of IEEE nuclear standards because these standards were written specifically to fit within this framework. Adapting standards that were written very broadly to fit the regulatory and standards framework of many different industries is only a slightly more difficult problem. The IEEE Software Engineering standards fit this class of broadly applicable standards. It is more difficult, however, to adapt for NRC use, standards that were specifically written for a different framework. Many of the IEC nuclear standards fall into this category.

IEC 880 is a case in point. At a minimum, the following framework dimensions must be addressed to integrate IEC 880 into the framework used in the U.S.:

- Regulatory authority — the regulatory requirements that the standard assumes overlay the standard’s requirements.
- System-level application — the type of system that the standard assumes the I&C system controls, e.g., aircraft flight control vs. reactor protection, LWR reactor protection vs. RBMK reactor protection.
- I&C system-level technical requirements — the I&C system technical requirements that the standard assumes are imposed by requirements of system-level standards.
- Program level quality assurance (QA) standards — the programmatic QA requirements that the standard assumes are imposed by program-level QA standards.
- Fundamental philosophy — what is being standardized, such as what is to be accomplished, how something is to be accomplished, or the configuration of system or component interfaces.
- Usage — how the standard is assumed to be used, e.g., to achieve regulatory compliance, to achieve interchangeability, or to establish contractual requirements.
- Scope — the scope of the standard relative to other standards in the framework. To minimize the potential for conflicting requirements and to allow for update of a standard independently from other standards, the amount of overlap in the scope of any two standards should be minimized.

Table 2 outlines the characteristics of IEC 880, the IEEE Nuclear Standards, and the IEEE Software Engineering Standards with respect to the above dimensions of a standards framework. It is clear that the framework of IEC 880 is both specific and different from the existing framework of U.S. nuclear standards. Section 3.7 discusses these differences in more detail. Additional information on the use of IEC 880 in a standards framework context is given in Joannou and Harauz, 1995.
Table 2. Comparison of Standards Frameworks

<table>
<thead>
<tr>
<th>Framework Dimension</th>
<th>IEC 880</th>
<th>IEEE Nuclear Standards</th>
<th>IEEE Software Engineering Standards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regulatory authority</td>
<td>Consistent with member states' nuclear regulations</td>
<td>10 CFR 50</td>
<td>Consistent with any application.</td>
</tr>
<tr>
<td>System-level application</td>
<td>Any type of nuclear power plant</td>
<td>ALWR (but really generic)</td>
<td>Consistent with any application. Must be integrated with system-level standards.</td>
</tr>
<tr>
<td>Program-level QA standards</td>
<td>IAEA 50-C-QA IAEA 50-SG-QA6</td>
<td>NQA-1</td>
<td></td>
</tr>
<tr>
<td>Philosophy</td>
<td>“How to”</td>
<td>“What to”</td>
<td>“What to”</td>
</tr>
<tr>
<td>Usage</td>
<td>Regulatory compliance</td>
<td>Regulatory compliance</td>
<td></td>
</tr>
<tr>
<td>Scope</td>
<td>System, computer, software, process</td>
<td>System, computer, software</td>
<td>Process or products --- varies by standard. The scope of individual standards is mostly independent.</td>
</tr>
</tbody>
</table>

Based on these considerations, it appears that the inclusion of international standards in a framework based on standards and regulations applicable to U.S. activities would involve significant investment to harmonize the recommendations of the international standards with positions already taken by the NRC. Furthermore, the use of international standards in concert with the IEEE nuclear and software engineering standards could, in many cases, cause continuing uncertainties regarding the mapping and consistency of information in the two sources.

However, benefits can be gained from the use of international standards. To maximize the benefit to U.S. industry, active U.S. industry participation in the IEC standards development process is desirable. Participation in international standards activities can help to ensure that developing or evolving international standards do not conflict in serious ways with practices in use in the U.S. It is beneficial for the NRC to monitor and participate in these activities in order to learn about potential conflicts as standards are evolving. This provides an opportunity to understand and plan for potential impacts as well as to provide feedback if appropriate. Also important to the NRC is the fact that products developed to international standards might be proposed for use in power plants within the regulatory purview of the NRC. In this case, it will be necessary for the NRC to understand how the requirements of the international standards relate to the standards framework used in the U.S. Although a regulatory guide will not be written for IEC 880, the analysis of this standard, done as part of the Standards Review task, has provided valuable information in this area. Finally, the review of IEEE working group activities indicates interest in international standards developments and, in some cases, there might be consideration of adopting applicable international standards rather than updating the IEEE standards.
3.0 DISCUSSION OF THE STANDARDS REVIEW TASK ANALYSES

The following sections describe the results of the analyses of the relationships among the selected standards. Each section addresses standards in one functional area, discusses working group activity and expected trends, and outlines the issues pertinent to the regulatory guides. The latter includes the planned approaches for dealing with these issues.

In addition to the specific items discussed below for the individual standards, there are several recurring themes that must be clarified for the nuclear safety context. Many standards, with respect to a variety of considerations, contain references to the consideration of budget and schedule. In the nuclear safety context, such considerations might be appropriate in specific situations, but must be consistent with regulation, e.g., they must never compromise safety. A second theme is that some standards refer to “critical” software although specific industry applications are not defined. For the regulatory guides, these references will be equated to safety-system software. Nuclear industry software developers may want to extend the definition of “critical” to other selected software because of the possible economic consequences of failures in that software. A third issue is that most standards include examples or recommendations in appendices. Unless specific items discussed in an appendix are required for the nuclear context, this material will not be included in the endorsement. A fourth theme is the occasional reference to pre-existing software products. The software engineering standards recognize the possibility of these products being used but provide only marginal guidance on how to qualify them. It will be noted, as appropriate, that specific regulatory guidance exists and should be consulted. Finally, the software engineering standards usually focus on the software development portions of the system life cycle; however, there are frequently general references to system issues such as the source of software requirements or acceptance of the software in the target environment. Where necessary, these references will be clarified in the context of regulation and nuclear industry standards.

The results of the Standards Review analyses are consistent with the SRP update work performed to date; however, the organization and presentation of some material, notably the Branch Technical Position on software process, has been influenced somewhat by these results. The schedule for the production of regulatory guides on software engineering issues and the schedule for the SRP update have been coordinated so that both the draft regulatory guides and the update of the SRP will be available for public comment at the same time. This coordination will ensure that the complete set of material is available during a single public comment period and will ensure that consistency and effectiveness of presentation will be maintained. The preliminary plans for the draft regulatory guides, outlined below, are also generally consistent with the discussions in Lawrence, 1993 and Seth et al., 1995.

3.1 Software Configuration Management (ANSI/IEEE Std 828 and ANSI/IEEE Std 1042)

3.1.1 Standard Status and Future Directions

As of this writing, there is no update activity on ANSI/IEEE Std 828-1990; an update of ANSI/IEEE Std 1042-1987 is currently being planned. The old IEEE Std 1033-1985 on the application of ANSI/IEEE Std 828 to nuclear power generating stations was withdrawn after the release of ANSI/IEEE Std 828-1990; this release specifically states that ANSI/IEEE Std 828-1990 does not supersede, revise, or amend industry-specific standards. Of interest are the requirements of NQA-1 and IEEE Std 7-4.3.2. The 828 working group has indicated interest in the draft of ISO/IEC 12207-2 on software configuration management.

Although there is currently no update activity on ANSI/IEEE Std 828-1990, there is a developing perception that configuration management may be accomplished in two distinguishable fashions —
classical configuration management, and modern, automated configuration management. Standards may eventually make this distinction.

3.1.2 Conflicts and Approaches

3.1.2.1 ANSI/IEEE Std 828-1990

Several areas of the regulatory guide for ANSI/IEEE Std 828-1990 are expected to contain exceptions, additions, or clarifications to the standard.

The most significant difference between ANSI/IEEE Std 828-1990 and other standards is that ANSI/IEEE Std 828-1990 does not define minimum requirements for configuration management of critical software, although software configuration management has been identified as a key element in the production of highly reliable software [Lawrence and Persons, 1994]. This issue will be addressed, most likely by pointing out that there are minimal requirements derived from regulation, such as 10 CFR Part 50 Appendix B, and by specifying minimum required activities. This is a difficult subject because ANSI/IEEE Std 828-1990 is a standard for writing configuration management plans, not for performing the configuration management function. It leaves blank the answers to the questions, “what should be under configuration management?” “how should it be done?” and “what minimum criteria does a configuration management function need to meet for safety-critical software?” Insofar as regulatory guides are “acceptable methods,” NRC staff opinion on what are acceptable minimum configuration management activities will be important. A related issue concerns the question of how changes originating in other activities, such as the handling of anomalies detected during V&V, relate to the change control function.

The definitions of terms in different standards often conflict with each other. “Baseline,” a crucial term underlying the concept of configuration management, is defined slightly differently by NQA-2a Subpart 2.7 and IEEE Std 1298. “Interface,” as used in “interface control,” is defined differently in IEEE Std 1016, ANSI/IEEE Std 830, and ANSI/IEEE Std 828. ANSI/IEEE Std 828 requires an “installation configuration audit,” while “functional configuration audits” and “physical configuration audits” are mentioned by ANSI/IEEE Std 1012 and IEEE Std 1028. The definition of configuration management is different between IEC 880 and most other standards reviewed. IEC 880 appears to limit configuration management to the maintenance of a software library. One of the activities basic to configuration management — change control — consists of some combination of anomaly reports, change requests, change approvals, change assessments, change implementation, change verification, and version release. The particular mix depends upon which standard is being referenced. ANSI/IEEE Std 828 requires the identification of configuration items that are under configuration control. At a minimum, any deliverable item must be a configuration item. IEEE Std 1028 mentions “software items” that must be identified, but never states whether “software items” are configuration items. ANSI/IEEE Std 1074 requires that, preceding or following certain life cycle activities, the configuration management activity should be “invoked.” This is taken to mean placing appropriate items under configuration control. ANSI/IEEE Std 828 defines “control points,” which may correspond to the ANSI/IEEE Std 1074 “invocations.” The draft regulatory guide can deal with all of these conflicts in definitions by noting the definition that the guide will use.

Many standards require documentation and record keeping. None of them suggest that the configuration management activity keep the records and cross-reference them to related configuration items under control. If something is important enough to be a configuration item, then other documentation bearing materially upon that configuration item should also be important enough to be under configuration control. This may require a staff interpretation, although the draft regulatory guide is only describing an “acceptable method.”

The question of the handling of vendor-supplied products is also important. It should be noted that contractually developed or qualified commercial software products that are used in or materially affect
the safety systems of nuclear power plants must be taken under control by a software configuration management program — operated by the using organization — that complies with ANSI/IEEE Std 828.

In the future, as automated configuration management tools become more common, the risk will increase that defects in the tools may lead to undetected configuration errors in products managed by the tools. The tools become, in effect, similar in impact to compilers and linkers. Consequently, a regulatory guide endorsing ANSI/IEEE Std 828-1990, to anticipate future developments, should discuss automated configuration management tools whose use figures significantly in an ANSI/IEEE Std 828-1990-compliant configuration management plan.

3.1.2.2 **ANSI/IEEE Std 1042-1987**

ANSI/IEEE Std 1042-1987 does not conflict with ANSI/IEEE Std 828-1990, because it is an elaboration of the latter (even though it was issued three years previously), and has the authority level of Guide as described in Section 2.2.1. As guidance, it describes the concepts and usage of versions, releases, and levels of configuration control and provides valuable information regarding the implementation of software configuration management. Many automated configuration management tools extant today use these concepts. The primary issues to be discussed in a regulatory guide on software configuration management pertain to ANSI/IEEE Std 828; clarifications and discussion only are anticipated for ANSI/IEEE Std 1042.

3.2 **Software Verification and Validation (ANSI/IEEE Std 1012 and IEEE Std 1028)**

3.2.1 **Standard Status and Future Directions**

As of this writing, the working group for ANSI/IEEE Std 1012-1986 is actively revising the standard. The focus is expected to shift from the software V&V plan to software V&V in general. This shift could put more emphasis on the process aspects of V&V activities rather than just describing the activities to be included in the plan. An attempt is also being made to ensure that the revision will be consistent with ISO/IEC 12207. Technical areas where significant changes might be made include the following:

- The use of four software integrity levels rather than the current critical/non-critical classification,
- Increased awareness of system and environmental interfaces, and
- Possibly providing some guidance on special topics, some of which may still be somewhat controversial; for example:
  - Risk management
  - Reusable software
  - Non-deterministic software
  - Independence.

IEEE Std 1028-1988 is nearing the end of a revision process. Much of the standard has been re-written to improve the standard; however, substantive changes to the requirements of the standard are not anticipated.

3.2.2 **Conflicts and Approaches**

3.2.2.1 **ANSI/IEEE Std 1012-1986**

The regulatory guide for ANSI/IEEE Std 1012-1986 is expected to contain exceptions, additions, or clarifications to the standard in a number of areas. These are summarized in the following paragraphs.
Configuration management is listed as an optional V&V task in ANSI/IEEE Std 1012-1986. The regulatory guide will describe configuration management as a necessary task, which may optionally be performed by V&V personnel. The regulatory guide is related to the regulatory guides for ANSI/IEEE Std 828-1990 and ANSI/IEEE Std 1074-1991 on this subject. The need for control of certain V&V materials under the configuration management system constitutes an additional requirement. The relationship between software configuration management and the V&V requirement for an anomaly process is of interest in the discussions on both V&V and software configuration management.

ANSI/IEEE Std 1012-1986 prescribes minimum tasks required for critical software. IEEE Std 7-4.3.2-1993 equates the term “critical” to “safety.” The regulatory guide will also take this position, referencing software included in I&C systems covered by Appendix B of 10 CFR 50. Safety is the primary criterion of interest to the NRC in a safety integrity level classification, such as that anticipated for the 1012 update. Safety-related software is expected to fall into the highest software integrity level of future classification schemes and, even then, additional requirements might apply.

ANSI/IEEE Std 1012-1986 is focused primarily on software issues, addressing system-level issues only briefly, such as in its discussion of Concept Phase V&V. The regulatory guide will note the integral relationship between software V&V and system issues, particularly with respect to safety. The importance of NQA-1-1994, IEEE Std 603-1991, and IEEE Std 7-4.3.2-1993 in this regard will be noted.

In its discussion of test plans for component and integration testing, ANSI/IEEE Std 1012-1986 lists measurement of software reliability as a criterion for determining if software elements correctly implement requirements. Although this measurement is recommended where feasible, exception must be taken to adopting it as a blanket requirement for all critical software since estimating software reliability to reasonable levels of confidence through statistical testing is not always practicable. Reliability requirements originate from system-level safety analyses. This subject is treated in Regulatory Guide 1.152, Revision 1, and the draft regulatory guide for ANSI/IEEE Std 1012-1986 will be consistent its guidance. The update of ANSI/IEEE Std 1012 is expected to address risk management, which is related to this subject; however, the regulatory guide will not mention this subject since the standard’s treatment of the subject has not yet been finalized and since the utility of ANSI/IEEE Std 1012 does not depend upon this subject.

Independence of V&V is not addressed in ANSI/IEEE Std 1012-1986 although it is planned for inclusion in the update. Independent verification is fundamental to nuclear industry practices and standards, as evidenced by 10 CFR Part 50 Appendix B (Design Control) as well as NQA-1-1994 and IEEE Std 7-4.3.2-1993. The regulatory guide will add a requirement regarding independence of V&V.

In its discussion of Operation & Maintenance V&V, ANSI/IEEE Std 1012-1986 provides guidance about applying the standard to the operation of software not developed to the standard. Exception is taken to these comments since the determination of acceptability and subsequent dedication of pre-existing software for use in safety-related systems in nuclear power plants is subject to specific guidance as described in the Branch Technical Position on the use of COTS software. It is likely that this exception will also be necessary for the anticipated update of ANSI/IEEE Std 1012.

Finally, a number of V&V tasks are described in ANSI/IEEE Std 1012-1986 as optional tasks. Of the tasks listed as optional, the regulatory guide will contain additions to require the following: various audits, configuration management (see above), installation and checkout testing, regression testing and analysis, test evaluation, and user documentation evaluation.

3.2.2.2 IEEE Std 1028-1988

IEEE Std 1028-1988 describes a process for performing software reviews and audits. It describes five generic classes of reviews: management reviews, technical reviews, inspections, walkthroughs, and
audits. Informative annexes suggest specific reviews that can be fitted into these five classes. These are based primarily on IEEE Std 730-1984 (which has since been revised to 730.1-1989).

IEEE Std 1028-1988 can be thought of as a “subroutine,” invoked procedurally whenever a review within the scope of ANSI/IEEE Std 1012-1986 is required, with the results handled according to the requirements of the software verification and validation plan (SVVP). It should be noted that the standard should also be followed for other reviews that may be required for quality assurance or management purposes. Since many technical reviews and audits will be specified in the SVVP, IEEE Std 1028 will be addressed in the same regulatory guide as ANSI/IEEE Std 1012-1986. The software developer should be expected to decide, based on regulatory requirements and the combined guidance of the standards framework in use, which reviews and audits will be carried out and how they match the five classes of IEEE Std 1028.

No major exceptions or additions to the body of IEEE Std 1028-1988 are anticipated. The phraseology used in IEEE Std 1028 differs from the typical style that employs the terms shall and should; therefore, a clarification will be necessary in the regulatory guide to indicate what content is considered to be required. The regulatory guide will emphasize that the annexes are informative and not part of the standard.

3.3 Software Requirements (ANSI/IEEE Std 830)

3.3.1 Standard Status and Future Directions

The most recent version of this standard, ANSI/IEEE Std 830-1993, does not differ substantially from the previous version. ANSI/IEEE Std 830 is a Recommended Practice as described in Section 2.2.1 and, as such, does not contain “shall.” A Guide is due out sometime in 1995. The working group is monitoring activity on IEEE Std 1498 on software processes as well as ANSI/IEEE Std 1074 on software life cycle processes.

3.3.2 Conflicts and Approaches

ANSI/IEEE Std 830-1993 is a Recommended Practice for the structure of software requirements specifications (SRSs). It is a product standard and, therefore, does not describe the process of creating an SRS. It does not discuss requirements analysis or verification of the SRS. Within the standards framework, IEEE Std 7-4.3.2-1993, ANSI/IEEE Std 1012-1986, and ANSI/IEEE Std 1074-1991 all relate to the questions of requirements analysis or verification. IEEE Std 7-4.3.2 addresses nuclear-industry–specific system issues, while ANSI/IEEE Std 1012-1986 addresses verification questions. The software life cycle processes standard notes the need for activities in both areas. These relationships will be discussed in the Discussion section of the regulatory guide for ANSI/IEEE Std 830-1993.

ANSI/IEEE Std 830 lists a number of attributes of an SRS that are desirable; the regulatory guide will probably make these attributes mandatory and will incorporate information regarding the attributes from appropriate nuclear industry sources. Seth et al., 1995 discusses issues associated with the definitive demonstration that such qualities have been achieved.

The Recommended Practice is meant to apply to all sorts of software; consequently, there is no language specific to real-time systems, to various operating modes of controlled processes, or to safety-critical applications. Clarifications regarding these issues might be necessary.

The Recommended Practice divides requirements into a number of categories: external interfaces, functions, performance, database, design constraints, and “other.” “Other” includes reliability, availability, security, maintainability, and portability. The description of this category is quite terse; for
example, reliability requirements are discussed in a single sentence. ("This should specify the factors required to establish the required reliability of the software system at time of delivery.") The language in the Recommended Practice is also limited in the Performance category, and appears to assume a large multi-user system, since it discusses number of terminals and number of simultaneous users. Additions or clarifications are possible regarding the performance, reliability, availability, and security categories.

Other aspects of ANSI/IEEE Std 830 to be considered include necessity of requirements, the degree to which it is appropriate to embed design information in requirements, and the documentation and control of interface requirements. ANSI/IEEE Std 830 refers to essential, conditional, and optional requirements; however, for safety-related software it is not clear that any requirements fall in the latter two categories. Design information derived from regulatory requirements may well be appropriate in software requirements.

3.4 Software Test Documentation (ANSI/IEEE Std 829)

3.4.1 Standard Status and Future Directions

ANSI/IEEE Std 829-1983 was re-affirmed in 1991. No update activity is currently underway.

3.4.2 Conflicts and Approaches

Documentation and evaluation of test results is required by Criterion XI of Appendix B to 10 CFR Part 50. The approach provided by ANSI/IEEE Std 829-1983 is an acceptable approach for structuring test documentation. Other structures exist, such as that found in the DOD standards. Therefore, it is quite possible that reasonable alternatives will be proposed. Since ANSI/IEEE Std 829-1983 does not prescribe which test documents to apply to specific testing phases (reference is made to this subject in Appendix B of the standard), the standard does not completely specify test documentation requirements. Recourse will be to 10 CFR 50 Appendix B and, in particular, to Criterion XI for provisions ensuring that prerequisites are met, that adequate test instrumentation is available and used, and that the test is performed under suitable environmental conditions. Criterion XI also requires the documentation and evaluation of test results. Any documentation necessary to meet the requirements of 10 CFR Part 50, Appendices A and B, must be noted as required information.

The needs for test documentation, its evaluation, and ensuring suitable environmental conditions imply that configuration management be applied to certain test materials, including documentation of the test configuration. This should be specified in the software configuration management plan.

A clarification is also appropriate to indicate that the various documents specified in ANSI/IEEE Std 829 can be packaged into other documents, provided that the constituent documents retain their identity. This addresses possible concerns about enabling efficient organization of the documentation.

3.5 Software Unit Testing (ANSI/IEEE Std 1008)

3.5.1 Standard Status and Future Directions

An effort update to ANSI/IEEE Std 1008-1987 is just beginning as of this writing. Areas to be reviewed include documentation, test strategy selection, coverage criteria, and special topics such as object-oriented techniques and artificial intelligence. The Software Testing Planning Group would like to create a higher-level testing standard that would encompass ANSI/IEEE Std 1008.
3.5.2 Conflicts and Approaches

ANSI/IEEE Std 1008-1987 is a standard that applies only to the software unit testing activity. No single IEEE standard or fabric of standards encompass all of the levels of testing, e.g., integration testing and system testing.

A degree of inconsistency exists between ANSI/IEEE Std 1012-1986 and ANSI/IEEE Std 1008-1987 regarding documentation. The latter requires only the ANSI/IEEE Std 829-1983 test design document and test summary report; however, it does require the activities, such as planning, that lead to most of the other documentation. The regulatory guide for ANSI/IEEE Std 1008-1987 will contain an addition to ensure that the appropriate documentation requirements of 10 CFR 50 Appendix B are satisfied in the unit testing activity.

In addressing completeness requirements, ANSI/IEEE Std 1008-1987 requires statement coverage for testing units coded in procedural languages. This is a weak coverage criterion, requiring only that each source language statement be executed as part of some test case. The regulatory guide will contain some discussion of coverage issues and require that the user specify and justify the coverage measures to be used. Seth et al., 1995 identifies coverage as a research area.

ANSI/IEEE Std 1008-1987 also discusses identifying features to be tested and, where complete testing is not practical, identifying the risk associated with features not tested. The regulatory guide will address this, noting that all safety features must be tested.

3.6 Software Life Cycle Processes (ANSI/IEEE Std 1074)

3.6.1 Standard Status and Future Directions

The balloting and re-work of an update to ANSI/IEEE Std 1074-1991 has been completed, and the document is being submitted to the IEEE Standards Board. It is expected to be available sometime in Fall 1995. The changes can be categorized as fine-tuning. Based on working group discussions about the current update, a substantive follow-on update effort is planned. The regulatory guide will be updated, if necessary, to reflect the 1995 version.

3.6.2 Conflicts and Approaches

ANSI/IEEE Std 1074-1991 is a process standard that describes activities to be incorporated into a software life cycle process. It describes a set of software processes spanning the life cycle from concept exploration to retirement. Each process consists of a set of activities, each of which is mandatory unless described in the standard as "if applicable." Some of the mandatory activities, such as "analyze risks" and "accept software in operational environment," touch on areas where additional guidance from system-level, industry standards must be considered. The regulatory guide for ANSI/IEEE Std 1074-1991 will clarify these links.

The specific requirements in the standard tend to be high-level and to reference various IEEE software engineering standards for additional information, although compliance with the referenced standards is not required for compliance with ANSI/IEEE Std 1074-1991. Hence, ANSI/IEEE Std 1074-1991 is largely consistent with the other IEEE software engineering standards; the need for exceptions, additions, or clarifications is focused on the relationship of the standard to system-level, industry standards. One specific instance is the need for software safety analyses such as the treatment of abnormal conditions and events in IEEE Std 7-4.3.2-1993. The subject is also discussed in Lawrence, 1993 and Seth et al., 1995. Software safety analyses will be additional mandatory activities imposed by the regulatory guide.
Other, more minor issues might also be addressed. In discussing the installation process, ANSI/IEEE Std 1074-1991 states that temporary “work-arounds” may be applied for problems identified and reported during installation. For safety-critical systems, these problems might constitute non-conformance and must be handled as such. ANSI/IEEE Std 1074-1991 defines the acceptance of software as the analysis of test summary information according to user acceptance plans. For safety-related software, acceptability includes the requirement that the software and its development processes comply with regulatory requirements. User acceptance plans should reflect these requirements. ANSI/IEEE Std 1074-1991 provides for user notification when a system is to be removed from active support or use. Regarding this requirement, it will be noted that the requirements of 10 CFR Part 21.51 for the maintenance of records apply for software used in safety systems. Finally, ANSI/IEEE Std 1074-1991 states that configuration management objectives are based on internal guidelines and contractual requirements. This will be amended to include regulatory requirements. Reference might be made to the regulatory guide for software configuration management.

3.7 Safety System Software (IEC 880)

3.7.1 Standard Status and Future Directions

A new version of the draft first supplement to IEC 880 is expected in the Fall of 1995. A second supplement is in development; its scope includes maintenance, security, on-line calculations and/or modeling, software architecture, communications links, languages, and software for hardware self-supervision. A draft version of the SC45A “chapeau” document has been produced. Experience with the use of IEC 880 is currently being collected and is expected to be evaluated in 1996.

3.7.2 Conflicts and Approaches

Because this standard is part of the international framework of standards, its approach and organization differ from the standards in the U.S. framework in two general areas: scope and prescriptiveness. IEC 880 addresses subjects that are variously addressed by IEEE Std 603, NQA-1, Subpart 2.7, ANSI/IEEE Std 828, ANSI/IEEE Std 1012, and ANSI/IEEE Std 830. Certain detailed practices are recommended or prescribed which U.S. standards discuss only in appendices or guides. An additional perspective on this issue is given in Joannou and Harauz, 1995. As a result of these differences and the large number of anticipated exceptions and additions, a regulatory guide for IEC 880 will not be produced as part of the statement of work. A regulatory guide for ANSI/IEEE Std 1074 (see above) will be added.

4.0 CONCLUSIONS

This review of individual standards and their relationships provided an opportunity not only to identify key issues that might be addressed in regulatory guides but also to view the standards from the perspective of coordinated sets of standards that would be applied to the development of software for nuclear applications. It became clear in this work that certain sets of standards (frameworks) emerged based on compatibility in factors such as scope, style, and approach. Standards within a framework relate to each other in a number of dimensions, all of which must be understood in order to make effective use of the standards. The importance of a thorough understanding of the standards framework associated with software development activities has been underscored by these results. The context for software development processes provided by this understanding is crucial for both the developer and reviewer. This view of standards relationships has also been of value in the related work on the update to the Standard Review Plan, in that its focus on a coordinated framework of standards has helped to ensure an effective, coordinated presentation of plans and Branch Technical Positions.
In addition to the standards framework in use in the U.S., there is a framework based on international standards. This framework differs from the U.S. framework in several ways, including the basis for the standards (IAEA Codes as opposed to NRC regulations) and the style and approach taken in the individual standards. Although these differences exist and pose difficulties for basing software processes on a combination of standards selected from each framework, there are several reasons for monitoring and participating in the development of international standards. The first of these is that, despite differences in style and approach, the technical information contained in the standards is beneficial and should be considered as an additional source for information regarding the conduct of software engineering activities. Participating in the development of these standards provides a mechanism for monitoring current content and future directions, and might help to prevent serious conflicts between frameworks. Second, it is likely that products developed in compliance with these standards will be proposed for use in applications within the U.S., which means that standards relationships must be understood. Third, in reviewing the planned future directions for some of the IEEE software engineering standards, it was learned that there is much interest in the international standards, and future revisions may migrate toward or simply adopt them.

One of the important results to emerge from this work is that the software engineering standards selected for regulatory guide endorsement (with ANSI/IEEE Std 1074-1991 added and IEC 880-1986 removed) are generally compatible with each other and with the nuclear industry standards in use in the U.S. The review of these standards yielded preliminary information regarding issues and approaches to be taken in the development of the regulatory guide drafts, as noted above. This information was the basis for the addition of ANSI/IEEE Std 1074 and the deletion of IEC 880 as standards for endorsement in this set of regulatory guides. It should be noted that IEC 880 is an important standard and products developed according to its guidance may be proposed for use in U.S. applications. In these cases, the results of the analyses on IEC 880 performed as part of this work will be helpful in the evaluation of such products.
REFERENCES


<table>
<thead>
<tr>
<th>Std. Org.</th>
<th>Num.</th>
<th>Title</th>
<th>Date</th>
<th>Authority</th>
<th>Criticality</th>
<th>Industry Focus</th>
<th>Life-Cycle (SW Development)</th>
<th>Special Areas</th>
<th>Scope Tailoring</th>
<th>SW Technology Orientation</th>
<th>System Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANSI</td>
<td>10.4</td>
<td>Guidelines for the Verification and Validation of Scientific and</td>
<td>1987</td>
<td>Guide</td>
<td>not specified</td>
<td>Nuclear Industry</td>
<td>Map reqs to LC used</td>
<td>S&amp;E Software; a posteriori V&amp;V</td>
<td>Level of effort; posterior applicability</td>
<td>Process</td>
<td>Software</td>
</tr>
<tr>
<td>ASME</td>
<td>NQA-</td>
<td>Quality Assurance Requirements for Nuclear Facility Applications</td>
<td>1994</td>
<td>Standard</td>
<td>not specified</td>
<td>Nuclear Industry</td>
<td>Map reqs to LC used (P2.7)</td>
<td>Overall QA including SOA</td>
<td>None; non-mandatory guidance in Part III</td>
<td>Process</td>
<td>System</td>
</tr>
<tr>
<td>IEC</td>
<td>880</td>
<td>Software for Computers in the Safety Systems of Nuclear Power Stations</td>
<td>1986</td>
<td>Standard</td>
<td>Safety</td>
<td>Nuclear Power Plants</td>
<td>Must be defined; phase criteria given</td>
<td>None</td>
<td>None; use in conjunction w/ Comp HW &amp; Sys Integr/ Sids</td>
<td>Process &amp; Product</td>
<td>Software/System</td>
</tr>
<tr>
<td>IEC</td>
<td>45A</td>
<td>Software for Computers Important to Safety for Nuclear Power Plants. First Supplement to IEC 880 (DRAFT)</td>
<td>1994 Draft</td>
<td>Draft</td>
<td>Important to Safety</td>
<td>Nuclear Power Plants</td>
<td>Not specified</td>
<td>GMF; Formal Methods; Tools; PESP; Safety-Related (IAEA)</td>
<td>None</td>
<td>Process &amp; Product</td>
<td>Software/System</td>
</tr>
<tr>
<td>IEC</td>
<td>828</td>
<td>Software Configuration Management Plans</td>
<td>1990</td>
<td>Standard</td>
<td>Critical &amp; Non-critical</td>
<td>General</td>
<td>Entire LC (critical SW)</td>
<td>PESP</td>
<td>Add reqs; format changes; omissions if n/a to project</td>
<td>Process</td>
<td>Software</td>
</tr>
<tr>
<td>IEEE</td>
<td>829</td>
<td>Software Test Documentation</td>
<td>1983</td>
<td>Standard</td>
<td>not specified</td>
<td>General</td>
<td>All phases of testing; integrate into LC</td>
<td>None</td>
<td>Classes of SW &amp; Required Documents</td>
<td>Process &amp; Product</td>
<td>Software</td>
</tr>
<tr>
<td>IEEE</td>
<td>830</td>
<td>Software Requirements Specifications</td>
<td>1993</td>
<td>Recommended Practice</td>
<td>not specified</td>
<td>General</td>
<td>Not specified</td>
<td>PESP</td>
<td>n/a (Note 2); may need to expand for embedded SW</td>
<td>Product</td>
<td>Software</td>
</tr>
<tr>
<td>IEEE</td>
<td>1008</td>
<td>Software Unit Testing</td>
<td>1987</td>
<td>Standard</td>
<td>not specified</td>
<td>General</td>
<td>Not specified</td>
<td>None</td>
<td>None</td>
<td>Process</td>
<td>Software</td>
</tr>
<tr>
<td>IEEE</td>
<td>1012</td>
<td>Software Verification &amp; Validation Plans</td>
<td>1986</td>
<td>Standard</td>
<td>Critical &amp; Non-critical (Note 1)</td>
<td>General</td>
<td>Map reqs to LC used</td>
<td>PESP notes</td>
<td>Critical; Add optional V&amp;V tasks</td>
<td>Process</td>
<td>Software</td>
</tr>
<tr>
<td>IEEE</td>
<td>1016</td>
<td>Software Design Descriptions</td>
<td>1987</td>
<td>Recommended Practice</td>
<td>Critical &amp; Non-critical</td>
<td>General</td>
<td>Not specified</td>
<td>None</td>
<td>n/a (Note 2)</td>
<td>Product</td>
<td>Software</td>
</tr>
<tr>
<td>Std. Org.</td>
<td>Num.</td>
<td>Title</td>
<td>Date</td>
<td>Authority</td>
<td>Criticality</td>
<td>Industry Focus</td>
<td>Special Areas</td>
<td>Scope Tailoring</td>
<td>SW Technology Orientation</td>
<td>System Level</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>----------------------------------------------------------------------</td>
<td>--------</td>
<td>-----------</td>
<td>-------------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>------------------------------------------------------</td>
<td>---------------------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>ANSI</td>
<td>10.4</td>
<td>Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry</td>
<td>1987</td>
<td>Guide</td>
<td>not specified</td>
<td>Nuclear Industry</td>
<td>Map reqs to LC used; S&amp;E Software; a posteriori V&amp;V</td>
<td>Level of effort; a posteriori applicability</td>
<td>Process</td>
<td>Software</td>
<td></td>
</tr>
<tr>
<td>IEEE</td>
<td>1028</td>
<td>Software Reviews &amp; Audits</td>
<td>1988</td>
<td>Standard</td>
<td>not specified</td>
<td>General</td>
<td>Not specified</td>
<td>None</td>
<td>Using Org; where, when, &amp; deviations</td>
<td>Process</td>
<td>Software</td>
</tr>
<tr>
<td>IEEE</td>
<td>1042</td>
<td>Guide to Software Configuration Management</td>
<td>1987</td>
<td>Guide</td>
<td>not specified</td>
<td>General</td>
<td>Not specified</td>
<td>None</td>
<td>n/a (Note 2); Template for critical, embedded SW</td>
<td>Process</td>
<td>Software</td>
</tr>
<tr>
<td>IEEE</td>
<td>1058</td>
<td>Software Project Management Plan</td>
<td>1987</td>
<td>Standard</td>
<td>Critical &amp; Non-critical</td>
<td>General</td>
<td>Not specified</td>
<td>None</td>
<td>Add sections may be added</td>
<td>Process</td>
<td>Software</td>
</tr>
<tr>
<td>IEEE</td>
<td>1063</td>
<td>Software User Documentation</td>
<td>1987</td>
<td>Standard</td>
<td>not specified</td>
<td>General</td>
<td>Not specified</td>
<td>None</td>
<td>Add sections may be added</td>
<td>Product</td>
<td>Software</td>
</tr>
<tr>
<td>IEEE</td>
<td>1074</td>
<td>Developing Software Life Cycle Processes</td>
<td>1991</td>
<td>Standard</td>
<td>Critical &amp; Non-critical</td>
<td>General</td>
<td>Not specified</td>
<td>Add sections may be added; e.g., for critical SW</td>
<td>Process</td>
<td>Software</td>
<td></td>
</tr>
<tr>
<td>IEEE</td>
<td>P730.</td>
<td>Software Quality Assurance Plan</td>
<td>1989</td>
<td>Standard</td>
<td>Critical</td>
<td>General</td>
<td>Not specified</td>
<td>None</td>
<td>Non-critical; subset may be used</td>
<td>Process</td>
<td>Software</td>
</tr>
<tr>
<td>IEEE</td>
<td>(NPE C) 1033</td>
<td>IEEE Recommended Practice for Application of IEEE 828 to Nuclear Power Generating Stations</td>
<td>1985</td>
<td>Recommended Practice (withdrawn?)</td>
<td>Safety</td>
<td>Nuclear Power Plants</td>
<td>Not specified</td>
<td>None</td>
<td>No</td>
<td>Process</td>
<td>System</td>
</tr>
<tr>
<td>Std. Org.</td>
<td>Num.</td>
<td>Title</td>
<td>Date</td>
<td>Authority</td>
<td>Criticality</td>
<td>Industry Focus</td>
<td>Life-Cycle (SW Development)</td>
<td>Special Areas</td>
<td>Scope Tailoring</td>
<td>SW Technology Orientation</td>
<td>System Level</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>----------------------------------------------------------------------</td>
<td>-------</td>
<td>-----------</td>
<td>-------------</td>
<td>----------------</td>
<td>----------------------------</td>
<td>-------------------------------</td>
<td>----------------</td>
<td>---------------------------</td>
<td>--------------</td>
</tr>
<tr>
<td>IEEE</td>
<td>7-4.3.2</td>
<td>Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations</td>
<td>1993</td>
<td>Standard</td>
<td>Safety</td>
<td>Nuclear Power Plants</td>
<td>Include HW; Map reqs to LC used (by ref to P2.7)</td>
<td>Nuclear Safety System Issues</td>
<td>No</td>
<td>Product &amp; Process</td>
<td>System</td>
</tr>
</tbody>
</table>

**NOTES**

- **Note 1**: SVVP applies to all SW. Minimum V&V tasks are specified for critical software.
- **Note 2**: Guides provide informative information and usually do not impose requirements beyond the standard to which they apply.