U.S. NUCLEAR REGULATORY COMMISSION

STANDARD REVIEW PLAN

BRANCH TECHNICAL POSITION 7-14

GUIDANCE ON SOFTWARE REVIEWS FOR DIGITAL COMPUTER-BASED INSTRUMENTATION AND CONTROL SYSTEMS

REVIEW RESPONSIBILITIES

Primary - Organization responsible for the review of instrumentation and controls

Secondary - None

Review Note: The revision numbers of Regulatory Guides (RG) and the years of endorsed industry standards referenced in this branch technical position (BTP) are centrally maintained in Standard Review Plan (SRP) Section 7.1-T (Table 7-1). Therefore, the individual revision numbers of RGs (except RG 1.97) and years of endorsed industry standards are not shown in this BTP. References to industry standards incorporated by reference into regulation (IEEE Std 279-1971 and IEEE Std 603-1991) and industry standards that are not endorsed by the agency do include the associated year in this BTP. See Table 7-1 to ensure that the appropriate RGs and endorsed industry standards are used for the review.

Draft Revision 6 – August 2015

A. BACKGROUND

The staff's acceptance of software for safety system functions is based upon: (1) confirmation that acceptable plans were prepared to control software development activities, (2) evidence that the plans were followed in an acceptable software life cycle, and (3) evidence that the process produced acceptable design outputs. This branch technical position (BTP) provides guidelines for evaluating software life cycle processes for digital computer-based instrumentation and control (I&C) systems. These guidelines are based on reviews of applicant or licensee submittals, Electric Power Research Institute (EPRI) requirements for advanced reactor designs, and the analysis of standards and practices documented in NUREG/CR-6101, "Software Reliability and Safety in Nuclear Reactor Protection Systems." The structure of the BTP is derived from the review process described in SRP, Appendix 7.0-A.

It is assumed that the software development plans, described in this BTP, are implemented within a Quality Assurance (QA) program that conforms to regulatory requirements. The plans are assumed to augment and supplement the processes and procedures in the QA program.

A.1 Regulatory Basis

The regulatory bases are found in SRP Appendix 7.1-A, and are summarized below.

1. Title 10 of the *Code of Federal Regulations* (10 CFR) 50.54(jj) and 10 CFR 50.55(i) require that Structures, systems, and components subject to the codes and standards in 10 CFR 50.55a must be designed, fabricated, erected, constructed, tested, and inspected to quality standards commensurate with the importance of the safety function to be performed.

2. 10 CFR 50.55a(h), "Protection and Safety Systems," requires compliance with the Institute of Electrical and Electronics Engineers (IEEE) Standard (Std) 603-1991, "IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations," and the correction sheet dated January 30, 1995. For nuclear power plants with construction permits issued before January 1, 1971, the applicant or licensee may elect to comply instead with their plant specific licensing basis. For nuclear power plants with construction permits issued between January 1, 1971 and May 13, 1999 the applicant or licensee may elect to comply instead with the requirements stated in IEEE Std 279-1971, "Criteria for Protection Systems for Nuclear Power Generating Stations." Clause 4.3 of IEEE Std 279-1971 states in part 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. Similar criteria for the quality of components are identified in IEEE Std 603-1991. Additional guidance on the application of IEEE
Std 279-1971 is provided in SRP Appendix 7.1-B. Additional guidance on the application of IEEE Std 603-1991 is provided in SRP Appendix 7.1-C.

3. 10 CFR Part 50, "Domestic Licensing of Production and Utilization Facilities," Appendix A, "General Design Criteria for Nuclear Power Plants," General Design Criterion (GDC) 1, "Quality Standards and Records," requires in part that systems and components important to safety be designed, fabricated, erected, and tested to quality standards commensurate with the importance of the safety functions to be performed. Where generally recognized codes and standards are used, they should be identified and evaluated to determine their applicability, adequacy, and sufficiency, and should be

supplemented or modified as necessary to ensure a quality product consistent with the required safety function.

4. GDC 21, “Protection System Reliability and Testability,” requires in part that protection systems be designed for high functional reliability commensurate with the safety function to be performed.

5. 10 CFR Part 50, Appendix B, “Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants,” Criterion III, “Design Control,” requires in part that quality standards be specified and that design control measures shall provide for verifying or checking the adequacy of design. Criterion V, “Instructions, Procedures, and Drawings,” requires in part that activities affecting quality shall be prescribed by “documented…procedures…of a type appropriate to the circumstances…” This BTP outlines such procedures for software. Criterion VI, “Document Control,” requires in part that “measures shall be established to control the issuance of documents…which prescribe all activities affecting quality…These measures shall assure documents, including changes, are reviewed for adequacy and approved for release by authorized personnel…” Criterion VII, “Control of Purchased Material, Equipment, and Services," addresses control of purchased material, equipment, and services. Further, Criterion XI, “Test Control,” requires in part that a test program be established to demonstrate that systems and components will perform satisfactorily in service.

A.2 Relevant Guidance

The relevant guidance is found in SRP Appendix 7.1-A, and is summarized below.

RG 1.28, “Quality Assurance Program Criteria (Design and Construction),” which endorses American Society of Mechanical Engineers (ASME) Nuclear Quality Assurance (NQA)-1, “Quality Assurance Requirements for Nuclear Facility Applications,” and the ASME NQA-1a, “Addenda to ASME NQA-1-2008, Quality Assurance Requirements for Nuclear Facility Applications,” and is subject to the provisions and modifications identified in the RG, provides an adequate basis for complying with the pertinent quality assurance requirements of Appendix B to 10 CFR Part 50.

RG 1.152, “Criteria for use of Computers in Safety Systems of Nuclear Power Plants,” which endorses IEEE Std 7-4.3.2, “IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations,” with a few noted exceptions, provides guidance for complying with requirements for safety systems that use digital computer systems. Additional guidance on the application of IEEE Std 7-4.3.2 is provided in SRP Appendix 7.1-D.

RG 1.168 endorses IEEE Std 1012, “IEEE Standard for Software Verification and Validation,” as providing methods acceptable to the NRC for meeting the regulatory requirements as they apply to verification and validation (V&V) of safety system software, subject to the exceptions listed. Further, it also endorses IEEE Std 1028, “IEEE Standard for Software Reviews and Audits,” as providing an approach acceptable to the staff for carrying out software reviews, inspections, walkthroughs, and audits, subject to the exceptions listed.
RG 1.169, “Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” endorses IEEE Std 828, “IEEE Standard for Configuration Management Plans,” as providing an acceptable approach for planning configuration management, subject to specific provisions identified in the RG.

RG 1.170, “Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” which endorses IEEE Std 829, “IEEE Standard for Software Test Documentation,” subject to the provisions and exceptions identified in the RG, identifies an acceptable method for satisfying test documentation requirements.

RG 1.171, “Software Unit Testing for Digital Computer Software used in Safety Systems of Nuclear Power Plants,” which endorses the American National Standards Institute (ANSI)/IEEE Std 1008, “IEEE Standard for Software Unit Testing,” subject to the provisions and exceptions identified in the RG, identifies an acceptable method for satisfying software unit testing requirements.

RG 1.172, “Software Requirements Specifications for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” which endorses IEEE Std 830, “IEEE Recommended Practice for Software Requirements Specifications,” subject to the provisions and exceptions identified in the RG, describes an acceptable approach for preparing software requirements specifications for safety system software.

RG 1.173, “Developing Software Life Cycle Processes for Digital Computer Software used in Safety Systems of Nuclear Power Plants,” which endorses IEEE Std 1074, “IEEE Standard for Developing Software Life Cycle Processes,” subject to the provisions and exceptions identified in the RG, as providing an approach acceptable to the NRC staff for meeting the regulatory requirements and guidance as they apply to development processes for safety system software.

NUREG/CR-6101, “Software Reliability and Safety in Nuclear Reactor Protection Systems.”

The information in this BTP is generally based on the regulatory basis and guidance identified above, supplemented as appropriate for attaining the required safety functions.

This BTP presents specific acceptance criteria for the elements of software reviews; however, important context information is found in the concepts contained in the referenced standards, reports, and RGs.

A.3 Definitions

Activity group - A collection of software life cycle activities, all of which are related to a specific life cycle topic.

Design output - Documents such as drawings and specifications, which define technical requirements of structures, systems, and components. For software, design outputs include the products of the development process that describe the end product that will be installed in the plant. The design outputs of a software development process include software requirements specifications, software design specifications, hardware and software architecture designs, code listings, system build documents, installation configuration tables, operations manuals, maintenance manuals, and training manuals.

Deterministic timing - Timing is deterministic if the time delay between stimulus and response has a guaranteed maximum and minimum.

Documentation - Information recorded about a specific life cycle activity. Documentation includes software life cycle design outputs and software life cycle process documentation. A

document may be in written or electronic format, and may contain text, illustrations, tables, computer files, program listings, binary images, and other forms of expression.

Plan - The document describing a specific implementation of a process (i.e., an instance of a process) that will be implemented.

Process - A series of actions, changes, or functions bringing about a result. A QA program is an example of a process definition.

Requirements Traceability Matrix (RTM) - An RTM shows every requirement, broken down in to sub-requirements as necessary, and what portion of the software requirement, software design description, actual code, and test requirement addresses that system requirement.

A.3.1 Definition of Software Planning Characteristics

The following paragraphs define the software planning characteristics important to safety system software. The definitions given are specific to software. The planning characteristics can be divided into three sets: management, implementation, and resource characteristics.

A.3.1.1 Definition of Management Characteristics

The management characteristics are those characteristics that are significant to the management of the project activities described in the planning document.

Purpose - A description of the reasons for the existence of the planning document, and the objectives which are to be satisfied by the planning document.

Organization - The organizational structure used to achieve the purpose of the planning document, including organizational boundaries and interfaces to other organizations.

Oversight - A specification of the methods used to oversee the work covered by the planning document.

Responsibilities - The duties of the organization covered by the planning document, and of the individuals within that organization.

Risks - The method used to identify, assess and manage risks that may interfere with achieving the purpose of the planning document.

Security - The methods used to protect the information created by or reviewed by the organization covered by the planning document from inadvertent or non-malicious alteration per RG 1.152. NRC published 10 CFR 73.54, “Protection of Digital Computer and Communication Systems and Networks,” to require licensees to develop cyber-security plans and programs to protect critical digital assets, including digital safety systems, from malicious cyber-attacks. RG 5.71, “Cyber Security Programs for Nuclear Facilities,” provides guidance to meet the requirements of 10 CFR 73.54.

A.3.1.2 Definition of Implementation Characteristics

The implementation characteristics are those characteristics of planning documents that describe the work necessary to achieve the purpose of the planning documents.

Measurement - A set of indicators used to determine the success or failure of the activities and tasks defined in the planning document.

Procedures - The documentation that describes the work necessary in order to achieve the purpose of the planning document.

Record keeping - Identification of: (1) the documentation required in order to document the work performed, (2) demonstrate that the purpose of the planning document has been achieved, and (3) the tasks necessary to store, handle, retain and ship the documentation.

Schedule - The time order of events necessary to achieve the purpose of the planning document, given either as absolute dates, ranges of dates, or offsets from other dates.

A.3.1.3 Definition of Resources Characteristics

The resources characteristics are those characteristics of planning documents that describe the material resources necessary to carry out the work defined in the planning document.

Budget - The financial resources necessary to carry out the work.

Methods/tools - The methods and techniques by which the work will be carried out, and the tools used to implement those methods.

Personnel - The numbers, qualification, and training of personnel required to carry out the work defined in the planning document.

Standards - The international, national, industry and company standards and guidelines to be followed in the work defined in the planning document.

A.3.2 Definitions of Functional and Process Characteristics

The following paragraphs define the software characteristics important to safety system software. The definitions given are specific to software. Software characteristics can be divided into two sets: functional characteristics and process characteristics. The first set includes those characteristics that directly relate to the actions that the safety system software must take, while the second includes those characteristics of the software development process that contribute to assurance that the software will perform the required actions. Both sets are important in safety system software. The sets, and the definitions of the characteristics, are listed below.

A.3.2.1 Definition of Functional Characteristics

Accuracy - The degree of freedom from error of sensor and operator input, the degree of exactness exhibited by an approximation or measurement, and the degree of freedom from error of actuator output.

Functionality - The operations which must be carried out by the software. Functions generally transform input information into output information. Inputs may be obtained from sensors, operators, other equipment, or other software. Outputs may be directed to actuators, operators, other equipment, or other software.

Reliability - The degree to which a software system or component operates without failure. This definition does not consider the consequences of failure, only the existence of failure.

Robustness - The ability of a software system or component to function correctly in the presence of invalid inputs or stressful environmental conditions. This includes the ability to function correctly despite some violation of the assumptions in its specification.

Safety - Those properties and characteristics of the software system that directly affect or interact with system safety considerations. The other characteristics discussed in this BTP are important contributors to the overall safety of the software-controlled safety system, but are primarily concerned with the internal operation of the software. The safety characteristic, however, is primarily concerned with the effect of the software on system hazards and the measures taken to control those hazards.

Security - The establishment of a Secure Development and Operational Environment (SDOE) for digital safety systems to: (1) to prevent undocumented, unneeded, and unwanted modifications and (2) to protect against a predictable set of undesirable acts (e.g., inadvertent operator actions or the undesirable behavior of connected systems) that could challenge the integrity, reliability, or functionality of a digital safety system during operations. RG 1.152 contains guidance for SDOE.

Timing - The ability of the software system to achieve its timing objectives within the hardware constraints imposed by the computing system being used.

A.3.2.2 Definition of Process Characteristics

Completeness - Those attributes of the planning documents, implementation process documents and design outputs that provide full implementation of the functions required of the software. The functions which the software is required to perform are derived from the general functional requirements of the safety system, and the assignment of functional requirements to the software in the overall system design. (See also the definition of “Complete” in Clause 4.3.3 of IEEE Std 830, which is endorsed by RG 1.172.)

Consistency - The degree of freedom from contradiction within a single document or among the different documents and components. There are two aspects to consistency. Internal consistency denotes the consistency within the different parts of a component for example, a software design output is internally consistent if no set of design elements are mutually contradictory. External consistency denotes the consistency between one component and another for example, software requirements and the resulting code are consistent with one another if there are no contradictions between the requirements and the code. (See also the definition of “Consistent” in Clause 4.3.4 of IEEE Std 830, which is endorsed by RG 1.172.)

Correctness - The degree to which a design output is free from faults in its specification, design, and implementation. There is considerable overlap between correctness properties and properties of other characteristics such as accuracy and completeness. (See also the definition of “Correct” in Clause 4.3.1 of IEEE Std 830, which is endorsed by RG 1.172.)

Style - The form and structure of a planning document, implementation process document or design output. Document style refers to the structure and form of a document. This has connotations of understandability, readability, and modifiability. Programming style refers to the programming language characteristics of the software and programming techniques which are mandated, encouraged, discouraged, or prohibited in a given implementation.

Traceability - The degree to which each element of one life cycle product can be traced forward to one or more elements of a successor life cycle product, and can be traced backwards to one or more elements of a predecessor life cycle product. Traceability is central to the production of complex systems to ensure all requirements are implemented, checked and tested. (See also the definition of "Traceable" in Clause 4.3.8 of IEEE Std 830, which is endorsed by RG 1.172.)

Unambiguity - The degree to which each element of a product, and of all elements taken together, have only one interpretation. (See also the definition of "Unambiguous" in Clause 4.3.2 of IEEE Std 830, which is endorsed by RG 1.172.)

Verifiability - The degree to which a software planning document, implementation process document or design output is stated or provided in such a way as to facilitate the establishment of verification criteria and the performance of analyses, reviews, or tests to determine whether those criteria have been met. (See also the definition of "Verifiable" in Clause 4.3.6 of IEEE Std 830, which is endorsed by RG 1.172.)

B. BRANCH TECHNICAL POSITION

B.1 Introduction

Digital I&C safety systems must be designed, fabricated, installed, and tested to quality standards commensurate with the importance of the safety functions to be performed. Implementation of an acceptable software life cycle provides the necessary software quality.

Digital I&C systems can share code, data transmission, data, and process equipment to a greater degree than analog systems. Although this sharing is the basis for many of the advantages of digital systems, it also raises a key concern: a design using shared data or code has the potential to propagate a common-cause or common-mode failure via software errors, thus defeating the redundancy achieved by the hardware architectural structure. Greater sharing of process equipment among functions within a channel, that is, the same processing equipment will be performing a number of different safety functions, increases the consequences of the failure of a single hardware module and reduces the amount of diversity available within a single safety channel.

Because of these concerns, the staff review of digital I&C systems emphasizes quality, diversity, and defense-in-depth as protection against common-cause failures within and between channels. Software quality is an important element in preventing the propagation of common-cause failures.

The development of safety system software should progress according to a formally defined life cycle. Many life cycles have been defined in the technical literature and in national and international standards. These differ in the definitions of life cycle activity groups and in the order in which life cycle activities are performed. The software developer should select and document the software life cycle, and specify the products that will be produced by that life

cycle. The software developer can be the applicant or licensee, the vendor, a company working on behalf of either, or a commercial software development company.

All software development life cycles share certain characteristics. The activities that will be performed can be grouped into a number of categories (termed activity groups here); the activity groups are common to all life cycles. Life cycle activities produce process documents and design outputs which can be reviewed and assessed. Further information on life cycles, activity groups, and document contents can be found in NUREG/CR-6101.

B.2 Information to be Reviewed

The information to be reviewed is subdivided into three topic areas: planning, listed in Subsection B.2.1; implementation, listed in Subsection B.2.2; and design outputs, listed in Subsection B.2.3.

B.2.1 Software Life Cycle Process Planning

The information to be reviewed may be contained in the following documents:

- Software Management Plan (SMP).
- Software Development Plan (SDP).
- Software Quality Assurance Plan (SQAP).
- Software Integration Plan (SIntP).
- Software Installation Plan (SInstP).
- Software Maintenance Plan (SMaintP).
- Software Training Plan (STrngP).
- Software Operations Plan (SOP).
- Software Safety Plan (SSP).
- Software Verification and Validation Plan (SVVP).
- Software Configuration Management Plan (SCMP).
- Software Test Plan (STP).

B.2.2 Software Life Cycle Process Implementation

The information to be reviewed may be contained in the following:

- Safety analyses.
- Verification and validation analysis and test reports.
- Configuration management reports.
- Testing Activities.

One or more sets of these reports should be available for each of the following activity groups:

- Requirements.
- Design.
- Implementation.
- Integration.
- Validation.
- Installation.

- Operations and maintenance.

B.2.3 Software Life Cycle Process Design Outputs

The information to be reviewed may be contained in the following:

- Software requirements specifications (SRS).
- Hardware and software architecture descriptions (SAD).
- Software design specifications (SDS).
- Code listings.
- Build documents.
- Installation configuration tables.
- Operations manuals.
- Maintenance manuals.
- Training manuals.

System requirements documents should also be examined to provide context for this review.

B.3 Acceptance Criteria

An appropriate set of life cycle activities is provided in RG 1.173, "Developing Software Life Cycle Processes for Digital Computer Software used in Safety Systems of Nuclear Power Plants," which endorses IEEE Std 1074, "Standard for Developing Life Cycle Processes."

Commercial-off-the-shelf software and software embedded in commercial-off-the-shelf components, such as meters, circuit breakers, or alarm modules should be appropriately evaluated to confirm that required characteristics are met. EPRI Topical Report TR-106439, "Guideline on Evaluation and Acceptance of Commercial Grade Digital Equipment for Nuclear Safety Applications," as approved by NRC's safety evaluation dated July 17, 1997, describes an acceptable method for performing this evaluation. NUREG/CR-6421, "A Proposed Acceptance Process for Commercial Off-the-Shelf (COTS) Software in Reactor Applications," provides additional background information. The guidelines of this BTP may be used as appropriate in assessing the software engineering processes used to develop commercial software. See the discussion of the commercial dedication of pre-developed software (PDS) in SRP Appendix 7.0-A.

Conformance to a RG can be partial or complete (unqualified). For example, "format conformance" for a plan would mean that the plan would be in the format specified. "Content conformance" means that all guidance is carried out as defined, but not necessarily presented in the format specified.

Complete conformance to a RG (i.e., acceptable methods) only requires the reviewer to determine if conformance is achieved and that the system is safe.

For partial or qualified conformance, the reviewer should determine the type of conformance, if the conformance is acceptable, if conformance was achieved, and finally if the system is safe. For example, an ambiguous claim like "the software life cycle processes were developed consistent with RG 1.173," does not actually identify which parts of the guidance in the RG were followed, and therefore the acceptability of the process is not clear. In order to determine the level of conformance, the reviewer should ask, in a request for additional information (RAI),

that each guidance followed be identified. Once the RAI is answered, then the reviewer can determine the acceptability of the defined process, and proceed with the review.

The acceptance criteria are subdivided into three areas: Subsection B.3.1, planning; Subsection B.3.2, implementation; and Subsection B.3.3, design outputs.

B.3.1 Acceptance Criteria for Planning

This subsection addresses acceptance criteria for planning activities. The acceptance criteria address specific software development planning activities and products. These products, when found to be acceptable, provide the reviewer with additional criteria for reviewing the processes and products of subsequent life cycle activities, as discussed in Subsections B.3.2 and B.3.3 below.

Safety system development is a complex set of activities implemented by various organizational entities. The plans described in this BTP are one representation of the required activities in terms of coherent and cohesive views, with defined inter-relationships, to be implemented by defined organizational entities. For example, it is understood that the software verification and validation plan (SVVP) defines all of the activities performed by the software verification and validation team (SVVT), and no activities that will be performed by another team. However, a Test Plan (TP) may describe all of the testing activities, even if portions are performed by different groups. Other sets of plans are permissible so long as each plan descries a coherent and cohesive view of the activities to be performed.

Plans may be organized in a layered fashion such that a higher level plan is implemented by lower level plans. For example, the TP could be a higher level plan that would be implemented by lower level plans, specific to particular teams (i.e., the SVVP includes the testing activities performed by the SVVT, and the software development plan includes the testing activities performed by the software development team, with testing activities being coordinated by the TP). Plans other than those defined in this BTP should still contain all of the required information and meet all of the acceptance criteria in this BTP.

Acceptance criteria are divided into three sets: management characteristics, implementation characteristics, and resource characteristics. Each of these is further divided into specific characteristics, as shown in the following table. Not all specific characteristics occur for every plan.

Management Characteristics	**Implementation Characteristics**	**Resource Characteristics**
Purpose	Measurement	Budget
Organization	Procedures	Methods/tools
Oversight	Record keeping	Personnel
Responsibilities	Schedule	Standards
Risks		
Security		

All planning documents should be evaluated for the following process characteristics: consistency, style, traceability, unambiguity and verifiability.

Consistency - Each plan should be internally consistent, and the complete set of plans should be mutually consistent.

Style - Plans should be documented so that they can be understood both by the users of the plan and by the reviewers.

Traceability - The software management plan should be traceable back to system management planning; the remaining software plans should be traceable back to the software management plan; and the various process implementation documents and the design outputs should be traceable back to the relevant plans.

Unambiguous - The set of plans should not be ambiguous.

Verifiable - It should be possible to verify that the plans have been followed during the software project. The review and assessment of the quality of the plans provide a means of judging the competency of the development organization and management.

It might be the case, particularly when the applicant or licensee is planning for future plants, that the software plans are created in stages as information becomes available. For example, budget and schedule information might not be available when the initial plans are created. This is acceptable, provided that the information is added to the plans prior to the time the information is needed to carry out the plans.

In general, the staff review is intended to verify that safety is maintained through the development of high quality software. The staff does not review the various development plans to ensure that the process is cost-effective or on schedule. The staff should assess whether there is adequate budget and sufficient schedule to support a high quality design effort and provide the time required for a careful and thorough V&V effort. The staff review should focus on those aspects which can impact the safety and quality of the resulting software design.

For future plants, it is possible that more than one vendor could be involved in the design and implementation of the plant digital I&C systems and the associated software. The planning documents and the associated implementation activities of the various vendors should be coordinated by the principal vendor responsible for the design of the plant project. Of particular importance are the interface requirements of the systems by the various vendors and the integration of the systems.

It is also possible that since applicants or licensees generally contract the hardware, software or system development to a vendor, there may be two sets of planning documentation, that of the applicant or licensee and that of the vendor. Appendix B to 10 CFR Part 50 states that licensees may delegate to others, such as contractors, agents, or consultants, the work of establishing and executing the QA program, or any part thereof, but that the licensee shall retain responsibility therefore, and that the authority and duties of persons and organizations performing activities affecting the safety-related functions of structures, systems, and components should be clearly established and delineated in writing.

This means that the applicant or licensee will need a method of assuring that an appropriate QA program is established and effectively executed and a method of verifying, such as by checking, auditing, and/or inspection, those activities affecting the safety-related functions have been correctly performed.

In the case where the applicant or licensee has contracted with a vendor, the two sets of planning documents should be reviewed together, to assure not only that the vendor preforms

the appropriate activities, but also that the applicant or licensee has in place a method of assuring those activities are carried out correctly.

Acceptance Criteria for Management Characteristics of Planning Documents

The generally applicable acceptance criteria for the management characteristic for the planning documents are provided below.

Purpose - Each plan should state the purpose of the plan, and should be reviewed in accordance with that purpose.

Organization - Each plan should identify the organizational entities responsible for implementing the plan.

Oversight - Specific references, which may include the V&V program and the QA program, should be made if appropriate.

Responsibilities - All planned activities are assigned to a responsible organizational entity.

Risks - Clause 5.3.6, "Software Project Risk management," of IEEE Std 7-4.3.2, "IEEE Standard Criteria for Digital Computers in Safety Systems on Nuclear Power Generating Stations," which is endorsed by RG 1.152, "Criteria for Use of Computers in Safety Systems of Nuclear Power Plants," contains guidance on risk management.

Security - RG 1.152, "Criteria for Use of Computers in Safety Systems of Nuclear Power Plants," provides guidance on the establishment of a SDOE. RG 1.168, Section C.7.c., "Secure Analysis," provides guidance on the SDOE related V&V activities.

Acceptance Criteria for Implementation Characteristics of Planning Documents

The generally applicable acceptance criteria for the implementation characteristics of the planning documents are provided below.

Measurement - Various guidance and standards exist for software related measurements. That guidance which is applicable to all plans is listed below. Guidance which is applicable to a specific plan is listed under that plan.

RG 1.152 endorses IEEE Std 7-4.3.2, which, in Clause 5.3.1.1 provides for consideration of software quality metrics, and identifies IEEE Std 1061-1998, "IEEE Standard for Software Quality Metrics Methodology."

RG 1.173 endorses IEEE Std 1074, subject to provisions listed. IEEE Std 1074, Clause A.1.1.4, "Define Metrics," Clause A.1.3.5, "Collect and Analyze Metric Data," and Clause A.5.1.1.2, "Process Improvement Reviews," contain an acceptable approach relating to metrics.

Procedures - See 10 CFR Part 50, Appendix B, V "Instructions, Procedures and Drawings."

Record keeping - See 10 CFR Part 50, Appendix B, XVII "Quality Assurance Records."

Schedule - This is important only in that the planning documents should show that this type of planning has been done.

Acceptance Criteria for Resources Characteristics of Planning Documents

The generally applicable acceptance criteria for resource characteristics for the planning document are provided below.

Budget - The review will not be making any determination of the adequacy of the budget, other than to determine that the V&V organization was not subject to budget restrictions causing a limit on the time and personnel required to perform adequate V&V.

Methods/tools - It is important to remember that if the output of any tool cannot be proven to be correct, such as may occur if the tool produces machine language software code, the tool itself should be developed or dedicated as safety-related, with all the attendant requirements.

Personnel - The number of personnel is highly dependent on the complexity of the project and the development schedule required. The numbers, qualification, and training of personnel performing the V&V should be assessed during the review. The V&V personnel should be at least as qualified as the personnel doing the design in order to be effective, and since effective V&V may take as much effort as the design, the total number of V&V personnel should be taken into consideration when evaluating the V&V effort.

Standards - The standards used need to be appropriate to meet regulatory requirements. If the standard has not previously been reviewed and approved (e.g., by a RG), additional time for the review of the acceptability of the standard will be needed.

There are two types of standards conformance, full and tailored. For full conformance, each provision in the standard would be met. For partial conformance, each provision, recommendation, and permission that is being followed should be identified, so that they can be followed.

The reviewer may also encounter approaches that use a standard directly or by incorporation of that standard into other procedures. It is typical that organizations that fully conform use the standard directly. For partial conformance, the organization may incorporate relevant aspects into procedures that are followed.

See also 10 CFR Part 50, Appendix A, General Design Criterion -1 “Quality Standards and Records.”

B.3.1.1 Acceptance Criteria for Software Management Plan

RG 1.173 endorses IEEE Std 1074, subject to provisions listed. IEEE Std 1074, Clause A.1.2.7, “Plan Project Management,” contains an acceptable approach to software project management.

NUREG/CR-6101, Section 3.1.1, “Software Project Management Plan,” and Section 4.1.1, “Software Project Management Plan,” contain relevant guidance. The SMP and Software Project Management Plan are considered to be the new and old names, respectively, for this plan.

Section 3.1.1 of NUREG/CR-6101, which was issued in June of 1993, identifies IEEE Std 1058.1-1987, "IEEE Standard for Software Project Management Plans," as one way of organizing the SMP and identifies the minimum topics to be described. IEEE Std 1058-1998, is the most current version. However, IEEE Std 1058-1998 has not been endorsed by a RG.

The SMP describes the management aspects of the software development project. The SMP should exhibit the management, implementation and resource characteristics listed below.

B.3.1.1.1 Management Characteristics of the SMP

The management characteristics that the SMP should exhibit include purpose, organization, oversight, responsibilities, and security.

Purpose should involve defining the intent of the software project in the SMP. The SMP should list the general functions the software will be expected to provide, and each of these functions should be traceable to the system requirements. In addition, the SMP should provide an overview of the system within which the software will reside. A general overview of the project should be provided. The assumptions upon which the project is based should be stated. The scope of work for the software project, and the product and process goals, should be discussed.

Organization involves a description of the software project planning organization. The SMP should describe the software project organizational structure, and describe the interfaces and boundaries between the project organization and other company organizations. Management reporting channels should be described. The methods by which subcontractors and suppliers will be managed should be described.

The SMP should ensure that the quality assurance organization, the software safety organization and the software V&V organization maintain independence from the development organization. In particular, the plan should ensure that these assurance organizations not report to the development organization, and not be subject to the financial control of the development organization.

Oversight involves specifying the strategy for managing the software project. Project priorities should be listed. A method should be described to monitor progress against the SMP and to document progress at regular intervals in progress reports. A method should exist to identify any deviations from the SMP in time to take corrective action.

Responsibilities concern definition of the duties of each member of the project's management and technical teams. The SMP should include a policy statement that the development personnel who produce each design output required by the SMP have the primary responsibility for the quality of that output.

Security means the establishment of a SDOE per RG 1.152.

B.3.1.1.2 Implementation Characteristics of the SMP

The implementation characteristics that the SMP should exhibit include measurement and procedures.

Measurement involves the definition of a set of management indicators which will be used to monitor and control the project. The SMP should require that data associated with project

management be systematically collected and analyzed to determine the effectiveness of project management. (See Clause 4.5.3.6, "Metrics collection plan," of IEEE Std 1058-1998.)

Procedures refer to a description of the process by which the project will be managed. The SMP should describe project priorities, project assumptions, and monitoring and control methods. It should describe the approach to be followed for recording the rationale for key decisions made in specifying, designing, implementing, procuring and assessing the software.

A list of all deliverable software, test software, support software and associated documentation should be included. Project management reviews should be specified. The means for performing corrective action and process improvement should be described. Management reports should be described, and reporting channels should be described. Periodic progress reports should be required.

The SMP should define the means by which the remaining plans will be produced. It should provide a means of managing externally and internally generated changes in any of the plans. The people responsible for reviewing the various project plans and any changes to those plans should be listed, by name or by title. A means should exist for generating changes to the plans and for evaluating suggested changes. The plans should be under configuration management control.

B.3.1.1.3 Resource Characteristics of the SMP

The resource characteristics that the SMP should exhibit include budget, methods/tools and personnel.

Budget consists of a project budget for all project activities. A means should exist to track and report resource expenditures. Sufficient resources should exist to carry out the defined tasks. The SMP should ensure that quality assurance budgets, safety budgets, and V&V budgets not be subject to expropriation by the software development organization, in order to maintain financial independence of these assurance activities.

Methods/tools involve a description of the means used to manage the project. The SMP should identify the methods, techniques and tools required to carry out the project management, including office equipment, computer hardware, and computer software.

Personnel concerns specification of the numbers, qualifications, training and types of personnel required to conduct the project. Personnel resources for each project phase should be listed.

Safety and V&V personnel should be competent in software engineering in order to ensure that software safety and software V&V are effectively implemented.

B.3.1.1.4 Review Guidance for the SMP

The purpose of the staff review of the SMP is to ensure that the management aspects of the software development project are such that high quality software will result. This necessitates a deliberate and careful development process. The staff will generally review the SMP at the initial stage of the software and hardware design implementation life cycle, however for reviewers of digital upgrades to existing reactors; the staff will generally review this plan after the software development is complete.

There are several management characteristics that are of particular interest. Since the software development will generally be done by a vendor and not by the applicant or licensee, the interface between the applicant or licensee and vendor, and the method by which the quality of the vendor effort will be judged is critical. An important aspect of the SMP is the relationship between the software development group and the groups which check the quality of the software development process and the software itself. Generally, these are the quality assurance organization, the software safety organization and the software V&V organization. It is important that these groups maintain independence from the development organization, by both organization and function.

It is important that oversight of the vendors exists and is effective. There should be oversight by the applicant or licensee. Software or system vendors may not be familiar with nuclear requirements or with specific plant requirements, and therefore one of the more important aspects is oversight by the applicant or licensee that is effective and meets 10 CFR Part 50, Appendix B.

When reviewing the duties of each member of the project's management and technical teams, check to ensure that the personnel responsible for various items have the experience or training to perform those duties.

When reviewing the SDOE guidance, the reviewer should determine that the methods used are consistent with RG 1.152, and that the methods are used effectively.

Many of the other characteristics of the SMP are of minimal concern to the safety of the system, and therefore of minimal concern to the reviewer. The management indicators used and the process by which the project will be managed are primarily of concern to the software development organization, and should be reviewed only to the extent that safety of the final product is maintained.

The same is true of the budget and personnel from the resource characteristics. The budget and number of personnel for the project is a trade-off with the length of time required, and therefore is of minimal concern. The adequacy of the budget and personnel for the quality assurance organization, the software safety organization and the software V&V organization is of interest, and should be reviewed to ensure that those groups have adequate resources to support a high quality design effort. This will require some judgment, and it may require a justification by the applicant or licensee or vendor. In addition, safety and V&V personnel should be competent in software engineering in order to ensure that software safety and software V&V are effectively implemented. A general rule of thumb is that the V&V personnel should be as least a qualified as the design personnel.

B.3.1.2 Software Development Plan

RG 1.173, "Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants," endorses IEEE Std 1074, "IEEE Standard for Developing Software Life Cycle Processes," subject to exceptions listed, as providing an approach acceptable to the staff, for meeting the regulatory requirements and guidance as they apply to development processes for safety system software.

RG 1.152, "Criteria for Use of Computers in Safety Systems of Nuclear Power Plants," which endorses IEEE Std 7-4.3.2, "IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations," subject to the provisions and exceptions, provides

guidance for complying with requirements for safety systems that use digital computer systems. Clause 5.3.1 of IEEE Std 7-4.3.2 contains guidance on software development.

NUREG/CR-6101, Section 3.1.6, “Software Development Plan,” and Section 4.1.6, “Software Development Plan,” contain guidance on SDPs.

The SDP describes the plan for technical project development. The SDP should exhibit the management, implementation and resource characteristics listed below.

B.3.1.2.1 Management Characteristics of the SDP

The management characteristics that the SDP should exhibit include purpose, organization, oversight, and risks.

Purpose involves a description of the objectives of each life cycle activity group and its context within the overall project.

Organization describes the software life cycle that will be used in the project. The life cycle should include uniquely identifiable development, verification and support processes with well-defined inputs and outputs. The life cycle model should be documented in the SDP.

Oversight involves specifying the strategy for managing the technical development effort. Project priorities should be listed. Required software quality factors should be identified and ordered by importance. A method should be provided to monitor progress against the SDP and to document progress at regular intervals in progress reports. A method should exist to identify any deviations from the SDP in time to take corrective action.

Risks involve the identification, assessment and management of project risks. The SDP should describe the method to be used for risk identification, assessment and management, with particular attention to risks that have the potential for compromising safety. The SDP should describe the method to be used to identify and assess the risk factors associated with product engineering, development environment and program constraints. It should describe the mechanisms for tracking the risk factors and implementing contingency plans. Risk factors that should be included include system risks, mechanical/electrical hardware integration, risks due to size and complexity of the product, the use of pre-developed software, cost and schedule, technological risk, and risks from program interfaces (maintenance, user, associate contractors, subcontractors, etc.). The SDP should identify key design and implementation issues, and the preliminary studies, simulation modeling, and prototyping required to resolve them.

B.3.1.2.2 Implementation Characteristics of the SDP

The implementation characteristics that the SDP should exhibit include measurement, procedures, and schedule.

Measurement concerns a set of indicators used to determine the success or failure of the technical aspects of the development process and the resulting design outputs. The SDP should require data associated with the technical development of the design outputs to be collected and analyzed to determine software quality. The error rate found during the development phases should be measured, recorded, analyzed and reported.

Procedures refer to the division of each life cycle activity into well-defined tasks. The inputs to each activity and each task should be provided, and the sources of those inputs should be identified. The conditions that must be satisfied before each activity can begin should be described. The outputs from each activity and each task should be provided, and the destination of those outputs should be identified. The SDP should include a review at the end of each life cycle activity. Reports on the technical development work should be described. RG 1.173 describes acceptable methods for defining the inputs and outputs of the life cycle activities.

Schedule concerns a project schedule. The SDP should identify key work packages, milestones and hold points. Sufficient intermediate milestones should be identified to avoid unexpected schedule delays. Reviews and audits should be included in the schedule as project milestones. The schedule should justify the time anticipated to complete each task. A single schedule that includes both management and technical activities is acceptable.

B.3.1.2.3 Resource Characteristics of the SDP

The resource characteristics that the SDP should exhibit include methods/tools and standards.

Methods/tools involve a description of the software development methods, techniques and tools to be used. The approach to be followed for reusing software should be described. The SDP should identify suitable facilities, tools and aids to facilitate the production, management and publication of appropriate and consistent documentation and for the development of the software. It should describe the software development environment, including software design aids, compilers, loaders, and subroutine libraries. The SDP should require that tools be qualified with a degree of rigor and level of detail appropriate to the safety significance of the software which is to be developed using the tools. Methods, techniques and tools that produce results that cannot be verified or that are not compatible with safety requirements should be prohibited, unless analysis shows that the alternative would be less safe.

Standards refers to a list the international, national, industry, and company standards and guidelines (including RGs) to be followed in the project. This should include software requirements standards, software design standards and software coding standards and internal standards and engineering and physical standards that form the basis for the plant safety analysis. The RGs listed in SRP Table 7-1 provide an acceptable list of standards.

B.3.1.2.4 Review Guidance for the SDP

The staff review of the software development is primarily intended to determine that use of the SDP results in a careful and deliberate development process which will result in high quality software, suitable for use in safety-related systems in nuclear power plants. The details on how this will be done may be found in other plans, such as the SVVP, SCMP, and so forth. The important aspect of the software development plan is the method to be used to make sure these other plans are being applied. This would generally include a provision for effective oversight, where the strategy for managing the technical development is specified. The reviewer should determine if the software development plan allows the applicant or licensee or vendor to adequately monitor the software development process, and that any deviations from the software development process will be discovered in time to take corrective action.

Risks that should be specifically reviewed are those associated with risks due to size and complexity of the product, and those associated with the use of pre-developed software.

Complexity of the product should be addressed by vendors. The reviewer should determine that the applicant or licensee also considered this risk. The use of commercial software and hardware may be attractive due to cost, schedule and availability, but there is some risk that a commercial grade dedication process will show the items to lack the quality necessary for use in safety-related systems in nuclear power plants.

The SDP should clearly state which tasks are a part of each life cycle, and state the life cycle inputs and outputs. The review, V&V of those outputs should be defined.

Under the Resource Characteristics, the methods and tools to be used should be evaluated. Of particular interest is the method by which the output of software tools, such as compilers or assemblers, will be verified to be correct. The criteria from IEEE Std 7-4.3.2 is that software tools should be used in a manner such that defects not detected by the software tool will be detected by V&V activities. If this is not possible, the tool itself should be safety-related.

The SDP should list the international, national, industry, and company standards and guidelines, including RGs, which will be followed. The reviewer needs to determine if these standards and guidelines have previously been approved by the staff, and if not, the standard needs to be reviewed to ensure that following that standard will result in meeting NRC requirements. Coding standards should be checked against the suggestions contained in NUREG/CR-6463, "Review Guidelines for Software Languages for Use in Nuclear Power Plant Safety Systems," and the applicant or licensee should be asked to justify any deviations.

B.3.1.3 Software Quality Assurance Plan (SQAP)

Typically a SQAP will be implemented under an NRC approved QA program, such as one deemed acceptable by RG 1.28, "Quality Assurance Program Criteria (Design and Construction)," which endorses ASME NQA-1, "Quality Assurance Requirements for Nuclear Facility Applications," and the ASME NQA-1a Addenda, "Addenda to ASME NQA-1-2008 Quality Assurance Requirements for Nuclear Facility Applications."

The SQAP shall conform to the requirements of 10 CFR Part 50, Appendix B, and the applicant's or licensee's overall QA program. The SQAP would typically: (1) identify which QA procedures are applicable to specific software processes; (2) identify particular methods chosen to implement QA procedural requirements; and (3) augment and supplement the QA program as needed for software.

Clause 5.3.1, "Software Development" of IEEE Std 7-4.3.2, "IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations," which is endorsed by RG 1.152, "Criteria for Use of computers in Safety Systems of Nuclear Power Plants," contains guidance on software quality assurance.

Clause 3.3, "Software Quality Management Process," of IEEE Std 1074, "IEEE Standard for Developing Software Lifecycle Processes," which is endorsed by RG 1.173, "Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems in Nuclear Power Plants," subject to the provisions and exceptions noted, provides guidance on software quality management.

NUREG/CR-6101, Section 3.1.2, "Software Quality Assurance Plan," and Section 4.1.2, "Software Quality Assurance Plan," contain guidance on SQAPs.

The SQAP should exhibit the management, implementation and resource characteristics listed below. The SQAP should conform to the requirements of 10 CFR Part 50, Appendix B, and the applicant's or licensee's overall quality assurance program.

B.3.1.3.1 Management Characteristics of the SQAP

The management characteristics that the SQAP should include are purpose, organization, responsibilities and security.

Purpose refers to a general description of the quality assurance process, and the goals of that process. The SQAP should list the general functions the software QA organization will be expected to perform and specific objectives for this project, if applicable.

Organization involves a description of the software QA organization. The SQAP should describe the boundaries between the software QA organization and other company organizations. Reporting channels should be described.

Responsibilities concern a definition of the responsibilities and authority of the software QA organization. The SQAP should require the software QA organization to assess and evaluate system safety, reliability and maintainability characteristics of the software.

Security means that the applicant's or licensee's QA group conducts periodic audits to determine the effectiveness of the SDOE per RG 1.152.

B.3.1.3.2 Implementation Characteristics of the SQAP

The implementation characteristics that the SQAP should exhibit include measurement, procedures, and record keeping.

Measurement consists of a set of indicators used to determine the success or failure of the software QA effort. The SQAP should require quality assurance data to be systematically collected and analyzed to determine software quality.

Procedures involve a description of the software QA procedures for the entire software life cycle. The SQAP should provide for QA participation in the assessment and review of project-specific standards, methods and tools. The SQAP should describe the methods, procedures and controls used to ensure that technical, quality and other requirements are accurately stated in project documentation. Procedures should exist to identify, track and resolve project conditions adverse to quality. The SQAP should ensure that traceability is maintained through all phases of the software life cycle. Required software quality factors (listed in Subsection B.3.3 below) should be identified. Software QA reports should be described.

The software QA organization should participate in formal reviews and audits of the software development activity. Required reviews and audits should be listed in the SQAP, including review documentation requirements, evaluation criteria, anomaly reporting, and anomaly resolution procedures.

RG 1.168, "Verification, Validation, Reviews and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants," which endorses IEEE Std 1028, "IEEE Standard for Software Reviews and Audits," describes acceptable methods for QA software reviews and audits.

Record keeping concerns a description of the software QA record keeping requirements and procedures. A list of the documents subject to software quality assurance oversight should be included. The SQAP should describe storage, handling, retention and shipping procedures for these documents and for project quality records. Document structures (such as an annotated table of contents) should be provided. The document control mechanism should be specified.

B.3.1.3.3 Resource Characteristics of the SQAP

The resource characteristics that the SQAP should exhibit include methods/tools and standards.

Methods/tools refer to a description of the means that will be used to accomplish the quality assurance function. The SQAP should identify suitable facilities, equipment, methods, techniques and tools to facilitate the performance of the QA work. Computer equipment and software used to perform the QA work should be specified.

Standards involve a method to ensure that approved standards, methods and tools are applied throughout the software life cycle. The SQAP should provide a method to establish and maintain the standards and methods for software QA, software V&V and software configuration management (CM).

B.3.1.3.4 Review Guidance for the SQAP

The SQAP is one of the more important plans which will be reviewed. The reviewer should determine not only that the SQAP exhibits the appropriate management, implementation and resource characteristics discussed above, but also that following the SQAP will result in high quality software that will perform the intended safety function. The staff's review samples the design process and products rather than performing a 100 percent check of every function and every line of code, to evaluate the effectiveness of the applicant or licensee or vendor QA and V&V efforts. The staff samples various items to determine that the applicant or licensee or vendor QA and V&V efforts were performed correctly. It is expected that the thread audit and other samples of the quality of the software product will not find any errors not already discovered and documented by either the QA organization or the V&V team. If one or more errors are found, this indicates a potential weakness in the effectiveness of the QA organization.

10 CFR Part 50, Appendix B, allows the licensee to delegate the work of establishing and executing the QA program, but the licensee shall retain responsibility therefore. If the applicant or licensee does delegate the QA program to the vendor, the reviewer should determine how the applicant or licensee retains the responsibility, and how the applicant or licensee determines that the quality of the software is sufficient.

The organization of the software QA organization should be checked to ensure that there is sufficient authority and organizational freedom, including sufficient independence from cost and schedule to ensure that the effectiveness of the QA organization is not compromised. IEEE Std 1028 can be used as guidance.

B.3.1.4 Software Integration Plan (SIntP)

RG 1.173 endorses IEEE Std 1074, subject to provisions listed. IEEE Std 1074 Clause A.1.2.8, "Plan Integration," contains an acceptable approach relating to planning for integration.

NUREG/CR-6101, Section 3.1.7, “Software Integration Plan,” and Section 4.1.7, “Software Integration Plan,” contain guidance on SIntPs.

The SIntP should exhibit the management, implementation, and resource characteristics listed below.

B.3.1.4.1 Management Characteristics of the SIntP

The management characteristics that the SIntP should exhibit include purpose, organization and responsibilities.

Purpose refers to a general description of the software integration process, the hardware/software integration process and the goals of those processes. The SintP should include a general description of the software integration process and of the hardware/software integration process.

Organization involves a description of the software integration organization. The SintP should describe the boundaries between the software integration organization and other company organizations. Reporting channels should be described. It is acceptable for the integration organization to report to the development organization, or to be part of the development organization.

Responsibilities involve a definition of the responsibilities and authority of the software integration organization.

B.3.1.4.2 Implementation Characteristics of the SIntP

The implementation characteristics that the SIntP should exhibit include measurement and procedures.

Measurement refers to a set of indicators used to determine the success or failure of the integration effort. The SIntP should require that data associated with the integration of the software, and of the hardware/software combination, be collected and analyzed to determine the adequacy of the integration effort. The error rate found during integration activities should be measured, recorded, analyzed and reported.

Procedures involve an integration strategy. The SIntP should include methods, procedures and controls for software integration, and for combined hardware/software integration, and, when multiple vendors are involved, systems integration. Integration of design outputs and reports should be described. The SIntP should require documentation describing the software integration tests to be performed, the hardware/software integration tests to be performed, the systems integration, and the expected results of those tests.

B.3.1.4.3 Resource Characteristics of the SIntP

The resource characteristics that the SIntP should exhibit include methods/tools.

Methods/tools refer to a description of the methods, techniques and tools that will be used to accomplish the integration function. The SIntP should require that integration tools be qualified with a degree of rigor and level of detail appropriate to the safety significance of the software which is to be created using the tools.

B.3.1.4.4 Review Guidance for the SIntP

The SIntP is not as critical as some of the other plans. It should be checked for the various management, implementation, and resource characteristics shown above, but there is no particular review guidance. The software integration organization is generally the same group as the software developers. The exception may be if there is more than one group of software developers, if some of the software is dedicated commercial grade or is reuse of previously developed software. In these instances, the methods, procedures and controls for software integration become more critical, and should be reviewed carefully. Since every instance will be somewhat different, the reviewer should exercise judgment concerning the integration review.

B.3.1.5 Software Installation Plan (SInstP)

RG 1.173 endorses IEEE Std 1074, subject to provisions listed. IEEE Std 1074, Clause A.1.2.4, "Plan Installation," contains an acceptable approach relating to planning for installation.

NUREG/CR-6101, Section 3.1.8, "Software Installation Plan," and Section 4.1.8, "Software Installation Plan," contain guidance on SInstPs.

The SInstP should exhibit the management, implementation, and resource characteristics listed below.

B.3.1.5.1 Management Characteristics of the SInstP

The management characteristics that the SInstP should exhibit include purpose, organization and responsibilities.

Purpose refers to a general description of the installation process, and the goals of that process. A general description of the environment (such as temperature, humidity, vibration, and rack space) within which the computer system and software system is qualified to operate should be included in the SInstP.

Organization concerns a description of the software installation organization. The SInstP should describe the boundaries between the software installation organization and the broader safety system installation organization. Reporting channels should be described. It is acceptable for the installation to be performed by the development organization or by the customer.

Responsibilities involve a definition of the responsibilities and authority of the software installation organization. If installation is performed by the customer, then the delineation of responsibility between the development organization and the customer should be defined in such a way that misunderstandings in communications between the two organizations are kept to a minimum.

B.3.1.5.2 Implementation Characteristics of the SInstP

The implementation characteristics that the SInstP should exhibit include measurement and procedures.

Measurement consists of a set of indicators used to determine the success or failure of the installation effort. The SInstP should require that data associated with the installation be

collected and analyzed. The error rate found during installation activities should be measured, recorded, analyzed and reported.

Procedures involve a description of the installation strategy. The SInstP should describe procedures for software installation, for combined hardware/software installation, and systems installation. The SInstP should describe the methods, procedures and controls used to ensure that the success or failure of the installation effort can be reliably determined. Checks should be required to ensure that the computer system is functional, that the sensors and actuators are functional, that all cards are present and installed in the correct slots, and that the communication system is correctly installed. A check should be required to ensure that the correct software versions are installed on the correct computers. Installation reports should be described. The SInstP should require that anomalies discovered during installation be reported to the developer and resolved prior to placing the software into operation. Either the SInstP, or the SVVP, should require adequate testing to provide confidence that the installed system will perform its safety function.

Plans for installation of software on installed systems in operating plants should recognize the need to declare all affected functions inoperable according to the plant's technical specifications before proceeding with installation, and to conduct appropriate return-to-service testing before declaring the modified function operable.

B.3.1.5.3 Resource Characteristics of the SInstP

The resource characteristics that the SInstP should exhibit include methods/tools.

Methods/tools involve a description of the methods, techniques and tools that will be used to accomplish the installation function. The SInstP should require that installation tools be qualified with a degree of rigor and level of detail appropriate to the safety significance of the software which is to be installed using the tools.

B.3.1.5.4 Review Guidance for the SInstP

The SInstP should be checked for the various management, implementation, and resource characteristics shown above, but there are no specific review guidelines. Since the software is being installed in hardware, the personnel performing this installation should be a mix of the software and hardware personnel. The critical part of the software installation is the system test (Note: per IEEE Std 1012, Final System testing is considered a V&V test, and is the responsibility of the V&V group).

There should be written and approved procedures for software installation, and for combined hardware/software installation, and systems installation. In a sufficiently complex system, these procedures may contain some errors, and one of the things a reviewer should monitor is how these errors are identified, corrected and documented. These corrections should be subject to the same quality and configuration control as the rest of the system.

B.3.1.6 Software Maintenance Plan (SMaintP)

Clause 5.4.2.3, "Maintenance of Commercial Dedication," of IEEE Std 7-4.3.2, "IEEE Standard Criteria for Digital Computers in Safety Systems on Nuclear Power Generating Stations," which is endorsed by RG 1.152, "Criteria for Use of Computers in Safety Systems of Nuclear Power

Plants," contains guidance on maintenance and configuration management for commercially dedicated items.

NUREG/CR-6101, Section 3.1.9, "Software Maintenance Plan," and Section 4.1.9, "Software Maintenance Plan," contain guidance on SMaintPs.

The SMaintP should exhibit the management, implementation and resource characteristics listed below.

B.3.1.6.1 Management Characteristics of the SMaintP

The management characteristics that the SMaintP should exhibit include purpose, organization, responsibilities, risks, and security.

Purpose refers to a general description of the software maintenance process and the goals of that process. The SMaintP should list the general functions that the software maintenance organization will be expected to perform, and provide general information on obtaining field trouble reports. Maintenance should be limited to the process of modifying a software design output to repair nonconforming items or to implement pre-planned actions necessary to maintain performance. Modifications to improve performance or other attributes, or to adapt the design outputs to a modified environment should be considered design changes.

Organization involves a description of the software maintenance organization. The SMaintP should describe the boundaries between the software maintenance organization and other company organizations. Reporting channels should be described. Formal communication channels between the maintenance organization and the customers using the software should be provided, so that incorrect behavior of the software during operation can be identified, isolated and corrected. This communication structure should provide assurance that software failures during operation will not be ignored.

Responsibilities concern a definition of the responsibilities and authority of the software maintenance organization.

Risks involve a description of the method used for software risk management during maintenance, with particular attention to risks that have the potential for compromising safety.

Security concerns a description of the methods to be used to prevent unintended functionality introduced into the digital safety systems per RG 1.152.

The SMaintP should identify the controls needed over maintenance activities and maintenance and test equipment to prevent unauthorized changes to hardware, software and system parameters. At a minimum, the potential for introducing unauthorized changes during repair, testing and calibration should be addressed.

B.3.1.6.2 Implementation Characteristics of the SMaintP

The implementation characteristics that the SMaintP should exhibit include measurement and procedures.

Measurement refers to a set of indicators used to determine the success or failure of the maintenance effort. The SMaintP should require that data associated with maintenance

activities be collected and analyzed to determine the effectiveness of the maintenance effort. The error rate found during maintenance activities should be measured, recorded, analyzed and reported.

Procedures involve a description of the maintenance strategy. These procedures may consist of two parts. The first part concerns the actions by the applicant or licensee to maintain the software. The second (optional) part, if the applicant or licensee relies on a software vendor to perform certain maintenance functions, consists of the associated vendor's procedures.

The SMaintP should include procedures for problem reporting, and for resolution of those problem reports. The problem reporting procedure should give time and date of occurrence, a brief description of the problem (including the state of the system at the beginning of the occurrence) and a description as to what was done to correct the problem. It should require that reported problems be evaluated to allow the identification of nonconforming items and the performance of corrective actions as described in Sections XV and XVI of 10 CFR Part 50, Appendix B.

The SMaintP should describe the process for identification, documentation, evaluation, segregation where practical, and disposition of nonconforming items, and for notification to affected organizations. Evaluation of nonconforming items and corrective actions should include as appropriate evaluation with respect to the requirements of 10 CFR 50.59, "Changes, Tests and Experiments," and reporting per the requirements of 10 CFR Part 21, "Reporting of Defects and Noncompliance." Operability determinations should be made as part of the SOP. Nonconformance to design requirements dispositioned "use-as-is" or "repair" should be subject to design control (including verification and validation, quality assurance, safety analysis, and configuration management) measures commensurate with those applied to the original design. The SMaintP should require that as-built records reflect any accepted deviations and justification for that acceptance.

Because any error in safety system software presents the potential for common-cause failure of redundant functions, the SMaintP should require timely evaluation of the effects of reported problems to support equipment operability determinations as required by plant technical specifications.

Periodic analysis and reporting of problems and their resolution should be required along with recommendations for improving operation. There should be a requirement for reporting what actions were taken regarding these recommendations.

B.3.1.6.3 Resource Characteristics of the SMaintP

The resource characteristics that the SMaintP should exhibit include methods/tools.

Methods/tools involve a description of the methods, techniques and tools that will be used to accomplish the maintenance function. The SMaintP should describe the facilities required to maintain the delivered software. It should list and describe the software, hardware and associated documentation required to maintain the delivered software. The SMaintP should require that maintenance tools be qualified with a degree of rigor and level of detail appropriate to the safety significance of the software which is to be created using the tools.

B.3.1.6.4 Review Guidance for the SMaintP

The SMaintP is important because software maintenance is often done after the system has been delivered, installed and accepted, and has been in use for a period of time. By this time, the software development team may have moved on to other projects or other jobs, and the knowledge of how the software works and what it does is limited to the documentation on the software. In addition, software maintenance is often done when the software has failed for some reason, and there may be pressure to quickly fix the problem, and get the system back on line. This may adversely affect the quality of the design, verification, validation and test, and documentation of the modification.

For these reasons, the reviewer should determine that the SMaintP requires the same careful and deliberate procedures that other plans require, and to make sure that management controls are in place to implement these procedures.

This SMaintP, together with the SCMP, define what records are kept on the software, and who controls those records. If software is to be modified, it is critical to ensure that the right version of the software undergoes that modification. The procedures for testing modifications are also critical. There have been many "fixes" which actually introduce more errors than they fix. The regression testing requirements should specify that all the acceptance tests originally performed or a carefully selected and justified subset of the acceptance tests be used to ensure that no new problem has been created. Make sure that the review process is required to determine that the proposed software maintenance is actually maintenance, and does not introduce new functions or other design changes.

The SMaintP should also specify that the personnel performing the maintenance are fully qualified. This generally means the personnel should be equally qualified as the original design team. The software tools should also be qualified, and should be identical to those used during the original design. The SmaintP should have some provisions for qualifying a new revision of the tools, if the original version of the tools is no longer available.

This entire process may be made more complex due to vendor/licensee interaction. It will often be the case that the software maintenance is done by the original system vendor, and then two maintenance plans will be reviewed, that of the vendor to actually perform the maintenance, and that of the applicant or licensee to review and approve the changes caused by the maintenance. The applicant or licensee should have appropriate measures, consistent with its QA plan to ensure that the required modification is needed, appropriate and correct. Keep in mind that 10 CFR Part 50, Appendix B allows the licensee to delegate the work, but the licensee retains the responsibility for safety and quality.

B.3.1.7 Software Training Plan (STrngP)

RG 1.173 endorses IEEE Std 1074, subject to provisions listed. IEEE Std 1074, Clause A.1.2.6, "Plan Training," contains an acceptable approach relating to planning for training.

NUREG/CR-6101, Section 3.1.10, "Software Training Plan," contains guidance on STrngPs.

The STrngP should exhibit the management, implementation, and resource characteristics listed below.

B.3.1.7.1 Management Characteristics of the STrngP

The management characteristics that the STrngP should exhibit include purpose, organization, and responsibilities.

Purpose involves a description of the means necessary to ensure that training needs of appropriate plant staff, including operators and I&C engineers and technicians, are fully achieved. The StrngP should include a general description of the training facilities. **Organization** concerns a description of the software training organization. The interfaces between the training organization and the project management organization should be described. Reporting channels should be described. Trainers should have the necessary knowledge of the software operation to ensure that trainees understand its operating and maintenance requirements.

Responsibilities involve a definition of the responsibilities and authority of the training organization and training by customers.

B.3.1.7.2 Implementation Characteristics of the STrngP

The implementation characteristics that the STrngP should exhibit include measurement and procedures.

Measurement concerns a set of indicators used to determine the success or failure of the training effort. The STrngP should require that training data be collected and analyzed to determine the effectiveness of the training effort. The trainee error rate found at the end of training activities should be measured, recorded, analyzed and reported.

Procedures involve a description of the training procedures. The STrngP should list any documentation required to support the training effort. The training program should be described. The STrngP should require that training be specific to different job functions. Training products and reports should be described. Reporting requirements should be specified.

B.3.1.7.3 Resource Characteristics of the STrngP

The resource characteristics that the STrngP should exhibit include methods/tools.

Methods/tools refer to a description of the methods, techniques and tools that will be used to accomplish the training function. Training should be carried out on a training system which is equivalent to the actual hardware/software system.

B.3.1.7.4 Review Guidance for the STrngP

The STrngP may be quite simple or very complex, depending on whether the original vendor or the applicant or licensee is doing the maintenance. If the applicant or licensee has contracted with the vendor to do the maintenance, the applicant or licensee personnel only need to know how to operate the digital equipment, and this is typically less complex than the knowledge required to maintain the equipment. An intermediate step is that the applicant or licensee personnel perform first level maintenance, determining which sub-unit, such as an individual PC board, is failed, replacing that sub-unit, and then sending the unit to the vendor for repair. The vendor may offer training in the operation of the equipment, and with some site specific

additional training, this may be sufficient for operation. Maintenance training is more complex, in particular software maintenance. Training provided by the vendor will typically show how to use the software tools used to generate original programs, but may not provide sufficient detail to allow software failure analysis and corrective actions. The applicant or licensee may, however, have a qualified engineering staff to be able to do software maintenance themselves.

For these reasons, the reviewer should keep in mind the organization responsible for performing the operation and maintenance of the system, and who will be performing the training when determining the adequacy of the STrngP.

One possible method of judging the quality of the training is for the reviewer to take the training. If the vendor offers training in design and operations of the equipment and time allows, it would be beneficial for the reviewer to take this training. This will not only allow the reviewer to assess the training, but will make the reviewer more familiar with the system and software tools, thereby making other review determinations easier.

B.3.1.8 Software Operations Plan (SOP)

The SOP should exhibit the management, implementation and resource characteristics listed below.

B.3.1.8.1 Management Characteristics of the SOP

The management characteristics that the SOP should exhibit include purpose, organization, responsibilities and security.

Purpose refers to a general description of the operation of the software. The SOP should include a general description of the functions that the software is to perform, and a general discussion of the means of carrying out those functions.

Organization involves a description of the organizational structure necessary to control the software operation. The SOP should specify operator interface stations and actions required to support operation.

Responsibilities concern a description of the responsibilities and authority of the operators.

Security involves a description of the appropriate controls to operate the software system. The SOP should identify the physical, logical, and administrative controls needed over operation activities to prevent unauthorized changes to hardware, software and system parameters per RG 1.152.

B.3.1.8.2 Implementation Characteristics of the SOP

The implementation characteristics that the SOP should exhibit include measurement and procedures.

Measurement concerns a set of indicators used to determine the success or failure of the operating procedures. The error rate found during operation activities should be measured, recorded, analyzed and reported.

Procedures involve a description of the procedures necessary to start, operate and stop the software system. The SOP should require a description of procedures for executing the software in all operating modes, and procedures for ensuring that the software state is consistent with the plant operating mode at all times. The SOP should require a description of backup procedures for data and code, and the intervals at which backup should occur. The SOP should require a list of error messages, giving a description of the error indication, the probable interpretation of the error indication, and steps to be taken to resolve the situation.

B.3.1.8.3 Resource Characteristics of the SOP

The resource characteristics that the SOP should exhibit include methods/tools.

Methods/tools involve a description of the methods, techniques and tools that will be used to operate the software system. The SOP should describe the facilities required to operate the delivered software. It should list and describe the software, hardware and associated documentation required to operate the delivered software.

B.3.1.8.4 Review Guidance for the SOP

The primary item to check when evaluating the SOP is completeness. The reviewer should determine if all the required items are covered by the SOP. The security of the system, and in particular, the means used to ensure that there are no unauthorized changes to hardware, software and system parameters per RG 1.152. The reviewer should use judgment to assess the acceptability of these plans. In general, the existence of such a plan shows that the applicant or licensee has considered the problem and is prepared to respond to the problem.

B.3.1.9 Software Safety Plan (SSP)

NUREG/CR-6101, Section 3.1.5 “Software Safety Plan,” and Section 4.1.5 “Software Safety Plan,” contain guidance on SSPs. (Note: Section 3.1.5 of NUREG/CR-6101, which was issued in June of 1993, is based on a Draft of IEEE Std 1228-1994. The latest revision of IEEE Std 1228-1994 (R 2002), “IEEE Standard for Software Safety Plans,” has not been endorsed by a RG.)

A SSP should plan the implementation of all of the necessary software safety analysis activities. Guidance for the appropriate software safety analysis activities are found in the following sources.

NUREG/CR-6101, Section 3.2.2, “Requirements Safety Analysis,” and Section 4.2.2, “Requirements Safety Analysis,” contain guidance on safety analysis activities.

NUREG/CR-6101, Section 3.3.3, “Software Design Safety Analysis,” and Section 4.3.3, “Design Safety Analysis,” contain guidance on safety analysis activities.

NUREG/CR-6101, Section 3.4.1, “Code Safety Analysis,” and Section 4.3.2, “Code Safety Analysis,” contain guidance on safety analysis activities.

NUREG/CR-6101, Section 3.5.2, “Integration Safety Analysis,” and Section 4.5.2, “Integration Safety Analysis,” contain guidance on safety analysis activities.

NUREG/CR-6101, Section 3.6.1, “Validation Safety Analysis,” and Section 4.6.1, “Validation Safety Analysis,” contain guidance on safety analysis activities.

NUREG/CR-6101, Section 3.7.5, “Installation Safety Analysis,” and Section 4.7.1, “Installation Safety Analysis,” contain guidance on safety analysis activities.

NUREG/CR-6101, Section 3.8, “Operations and Maintenance Activities - Change Safety Analysis,” contains guidance on safety analysis activities.

RG 1.173, Section C.3, “Software Safety Analyses,” contains guidance on safety analysis activities.

The SSP should exhibit the management, implementation and resource characteristics listed below. The SSP should conform to the requirements of the design basis for the software applications involved.

B.3.1.9.1 Management Characteristics of the SSP

The management characteristics that the SSP should exhibit include purpose, organization, responsibilities and risks.

Purpose refers to a specification of the purpose and scope of the software safety activities. The SSP should provide a general description of the software safety effort, and the intended interactions between the software safety organization and the general system safety organization.

Organization involves a description of the software safety organization. The SSP should describe the boundaries and interfaces between the software safety organization and other company organizations. It should show how the software safety activities are integrated with the system safety activities, how the software safety activities are coordinated with the development activities, and the interactions between the software safety organization and the software V&V organization. The SSP should designate a single safety officer that has clear responsibility for the safety qualities of the software being constructed.

Responsibilities involve a definition of the responsibilities and authority of the software safety organization. The SSP should specify the person or group responsible for each software safety task. A designated safety officer should have clear authority for enforcing safety requirements in the software requirements specification, the design, and the implementation of the software. The safety officer should have the authority to reject the use of pre-developed software if the software cannot be shown to be adequately safe or if, in using a tool, it cannot be shown that the tool will not impact the safety of the final software system. The SSP should require that safety personnel be aware of the safety implications of hardware, software and interfaces between them.

Risks involve a description of the methods to be used to reduce safety risks caused by software failures to an acceptable level. The SSP should describe the method to be used to ensure that hazards which software is expected to control are resolved in an acceptable manner. The SSP should include a requirement that a safety analysis be performed and documented on each of the principal design documents: requirements, design descriptions, and source code. Hazards, including abnormal events and conditions and malicious modifications, should be analyzed and documented. Hazard reduction efforts should be documented.

B.3.1.9.2 Implementation Characteristics of the SSP

The implementation characteristics that the SSP should exhibit include measurement and procedures.

Measurement concerns a set of indicators used to determine the success or failure of the software safety effort. The SSP should require that software safety data be systematically collected and analyzed to determine the effectiveness of the software safety effort.

Procedures involve a description of the software safety strategy. The SSP should describe the management of the software safety activities within the development organization. It should provide procedures for resolving safety issues. The SSP should require that problems encountered in implementing the safety program be brought to the attention of the project manager. A procedure should exist for assuring resolution of identified unacceptable risks. The SSP should describe methods to be used to implement each safety task. A method should exist to identify hazards caused by software, and to identify hazards whose resolution will be under the control of software.

The SSP should require that appropriate safety requirements be included in the software requirements specification. It should define the safety-related activities to be carried out for each set of life cycle activities, from requirements through operation and maintenance. The SSP should identify all documentation required for the proper and safe operation of the software. Procedures should require monitoring the software safety function performance during operation of the system.

The SSP should require that hazards identified by plant safety analysis, system safety analysis and security vulnerability assessment be traceable to the software safety analysis whenever these hazards can affect software operability or whenever software has a role in controlling the hazard.

B.3.1.9.3 Resource Characteristics of the SSP

The resource characteristics that the SSP should exhibit include methods/tools and standards.

Methods/tools refer to a description of the methods and tools used to carry out the safety activities. The SSP should specify a process for selecting tools. It should describe a method for preventing the inadvertent introduction of hazards by the use of project tools.

Standards involve a list of the international, national, industry and company standards and guidelines to be followed by the safety organization.

B.3.1.9.4 Review Guidance for the SSP

It is important for the reviewer to evaluate the completeness of the SSP and the approaches for how the applicant/licensee will handle the various issues. It is also likely, as mentioned above, that there may not be a separate safety plan; the concepts of software safety may be included in the SMP or some other plan. As long as the concepts mentioned above are addressed, this is acceptable. There should be a group which specifically considers the safety issues of the digital system to determine the acceptability of the system. The safety organization should consider the security risk as well as the risk to the plant if the digital system malfunctions. The reviewer should assess whether the proper risks were considered, that the applicant/licensee addressed

these risks in an appropriate manner, and stayed consistent with the software safety strategy. Where possible the reviewer should assess the adequacy of the risk evaluations. The final result of the review is that the reviewer should assure that the various software safety activities will resolve the safety issues.

B.3.1.10 Software Verification and Validation Plan (SVVP)

A SVVP should plan the implementation of all of the necessary V&V activities. Guidance for the appropriate V&V activities is found in the following sources.

RG 1.152, Section C.2, "Secure Development and Operational Environment for the Protection of Digital Safety Systems," contains guidance for testing the SDODE design features.

RG 1.168 endorses IEEE Std 1012, "IEEE Standard for Software Verification and Validation," as providing methods acceptable to the staff for meeting the regulatory requirements as they apply to V&V of safety system software, subject to the exceptions listed.

RG 1.168 also endorses IEEE Std 1028, "IEEE Standard for Software Reviews and Audits," as providing an approach acceptable to the staff for carrying out software reviews, inspections, walkthroughs and audits, subject to the exceptions listed.

RG 1.168, Section C.7.c, "Secure Analysis," states that the NRC staff considers the assessment of the secure operational environment design features of safety system software to be part of the minimum set of software V&V activities.

NUREG/CR-6101, Section 3.1.4, "Software Verification and Validation Plan," and Section 4.1.4, "Software Verification and Validation Plan," contain guidance on SVVPs.

The SVVP should exhibit the management, implementation and resource characteristics listed below.

B.3.1.10.1 Management Characteristics of the SVVP

The management characteristics that the SVVP should exhibit include purpose, organization, oversight, responsibilities, and risks.

Purpose refers to a definition of the purpose and scope of the software V&V activities. The SVVP should include a general description of the software V&V process.

Organization involves a description of the V&V organization. The SVVP should describe the boundaries and interfaces between the V&V organization and other company organizations. Reporting channels should be described. The relationship among the different V&V tasks should be specified. The SVVP should require that the V&V organization be independent of the development organization. It should require that formal communication between the V&V and design organizations be documented.

Oversight involves a description of the software V&V organization. The SVVP should describe the boundaries and interfaces between the software V&V organization and other company organizations. It should show how the software V&V activities are integrated with the system V&V activities, how the software V&V activities are coordinated with the development activities,

and the interactions between the software V&V organization and the software safety organization.

Responsibilities concern a definition of the responsibilities and authority of the software V&V organization. The SVVP should specify the person or group responsible for the successful completion of each V&V task. It should specify the person with authority to approve the successful completion of each V&V task. It should specify the person with authority to approve the release of the reviewed and tested software design outputs.

Risks involve specification of the methods used to identify and manage risks associated with the V&V process. The SVVP should specify a method for evaluating the risk to safety associated with each software item. It should describe a method for identifying the risk associated with each V&V task. A contingency plan should be included to identify risk factors that may cause the V&V task to fail to perform its functions, and to recover from any such failure.

B.3.1.10.2 Implementation Characteristics of the SVVP

The implementation characteristics that the SVVP should exhibit include measurement and procedures.

Measurement consists of a set of indicators used to determine the success or failure of the software V&V effort. The SVVP should specify the criteria to be used to verify the completion of each V&V task. Evaluation criteria should be provided for test plans, test specifications, test procedures and test cases. Evaluation criteria should be provided for review plans, review specifications and review procedures. The SVVP should require that V&V analysis, review and testing data be systematically collected and analyzed to determine the effectiveness of the V&V effort. The error rate found during software reviews and software testing should be measured, recorded, analyzed and reported.

Procedures involve a description of the software review and testing strategy. The SVVP should describe the management of the software V&V activities. It should specify the V&V tasks which will be carried out, including the planning assumptions for each task. It should establish the procedures and methods by which each V&V task will be performed, including the activities required to evaluate each software design output and each development activity in order to demonstrate that the system and software requirements have been met. It should establish procedures to ensure that systems in which errors are detected are appropriately analyzed, reported, corrected and reassessed. The SVVP should provide procedures for evaluating the risks associated with each project development activity. It should include a procedure for evaluating the effect of proposed software changes on planned reviews and tests.

Anomaly reports should be generated and disseminated. A method should be specified for resolving discrepancies identified during the verification of each V&V task. Procedures should be specified for selecting test cases, and for software review activities.

The SVVP should describe V&V reporting requirements. It should require that reports document all V&V activities, including the personnel conducting the activities, procedures and results. This includes review documentation requirements, evaluation criteria, error reporting, and anomaly resolution procedures. V&V reports should summarize the positive practices and findings as well as negative practices and findings. The reports should summarize the actions performed and the methods and tools used.

The SVVP should include a description of all required testing plans, specifications, procedures and cases. This includes unit testing, integration (subsystem) testing, system validation testing, installation (acceptance) testing, and the regression testing of modifications. The description should also include test documentation requirements, readiness and evaluation criteria, error reporting, and anomaly resolution procedures. Testing documentation should include test item descriptions, test data, test logs, the identities of testers, types of observations, results and acceptability, and actions taken in connection with any deficiencies. Test case documentation should specify expected results and actual results.

RG 1.170, "Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants," which endorses IEEE Std 829, "IEEE Standard for Software Test Documentation," describes acceptable methods for documenting test plans, test specifications, test procedures, test cases, and test reports.

RG 1.171, "Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants," which endorses ANSI/IEEE Std 1008, "IEEE Standard for Software Unit Testing," describes acceptable methods for performing unit tests.

B.3.1.10.3 Resource Characteristics for the SVVP

The resource characteristics that the SVVP should exhibit include methods/tools and standards.

Methods/tools involve a description of the methods, equipment, instrumentation and tools used to carry out each V&V task. Test methods should be specified for unit, integration, validation, installation and regression testing. The SVVP should specify a process for selecting tools. The hardware and software environment within which the V&V tools are to be applied and any necessary controls should be described.

Standards refer to a list of the international, national, industry and company standards and guidelines to be followed by the V&V organization.

B.3.1.10.4 Review Guidance for the SVVP

The SVVP is a key document among the various plans reviewed. The licensee/applicant or vendor is expected to develop and implement a high quality process to ensure that the resultant software is of high quality. A V&V effort that is sufficiently disciplined and rigorous is essential to demonstration of a high quality software development process. For this reason, the staff confidence that the V&V effort will identify and solve the problems which could detract from a high quality design effort will require the careful review of the SVVP, as well as the various V&V reports.

One of the most critical items in the SVVP is the independence of the V&V organization. The V&V team should be independent in management, schedule and finance (per RG 1.168). The reviewer should check that the V&V team is independent. V&V personnel should not be subject to scheduling constraints or to pressure from the software designers or project managers for reports or review effort, and the V&V team should report to a level of management which is not exerting direct pressure for a favorable V&V report. The staff reviewer should determine if the V&V effort is sufficiently independent to adequately perform the tasks without bowing to schedule and financial pressure.

A second important assessment the staff reviewer should make is on the number and quality of the V&V personnel. There is no specific requirement for the number of V&V personnel, but as an industry rule of thumb, it takes as much effort for a good V&V process as is required for the original design effort. This would indicate that if the V&V team is to keep up with the design team, there should be rough parity between the two groups in the number and skills of the personnel assigned. If the design group significantly outnumbers the V&V group, either the V&V effort will fall behind, or the V&V effort will not perform all the items required. Since there is no numeric requirement, this would be an indication to the staff reviewer to look closely at the outputs from the V&V team to see if the V&V quality is acceptable, or if some functions are not being performed.

The quality of the V&V personnel is also important. If a V&V engineer is to judge the output of a software design engineer, the V&V engineer should be qualified to understand the process, the technology, and the software. If the V&V engineer is not qualified, the V&V effort may not be effective. The reviewer can either ask for the qualifications and resumes of the V&V personnel, or if sufficiently experienced, interview the V&V personnel to gauge their knowledge and experience.

The reviewer should check that the SVVP requires the results of the V&V effort to be fully and carefully documented, and that each of the discrepancies be documented in a report, and those discrepancies be resolved, with the resolution documented, tested and accepted by the V&V organization. A significant contributor to problems found in final products has resulted from fixes to problems, where the fix itself did not go through the V&V process, was not properly tested, and therefore the additional problems created by the fix were not found.

The SVVP should describe the V&V reporting requirements. It should require that reports document all V&V activities, including the personnel conducting the activities, procedures and results. This includes review documentation requirements, evaluation criteria, error reporting, and anomaly resolution procedures. V&V reports should summarize the positive practices and findings as well as negative practices and findings. The reports should summarize the actions performed and the methods and tools used.

B.3.1.11 Software Configuration Management Plan (SCMP)

RG 1.173 endorses IEEE Std 1074, subject to provisions listed. IEEE Std 1074 Clause A.1.2.2, “Plan Configuration Management,” contains an acceptable approach relating to planning configuration management.

RG 1.169, “Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” endorses IEEE Std 828, “IEEE Standard for Configuration Management Plans,” as providing an acceptable approach for planning configuration management.

Clause 5.3.5, “Software configuration management,” of IEEE Std 7-4.3.2, “IEEE Standard Criteria for Digital Computers in Safety Systems on Nuclear Power Generating Stations,” which is endorsed by RG 1.152, “Criteria for Use of Computers in Safety Systems of Nuclear Power Plants,” also contains guidance on software configuration management.

Clause 5.4.2.1.3, “Establish configuration management controls,” of IEEE Std 7-4.3.2, “IEEE Standard Criteria for Digital Computers in Safety Systems on Nuclear Power Generating Stations,” which is endorsed by RG 1.152, “Criteria for Use of Computers in Safety Systems of

Nuclear Power Plants," contains guidance on software configuration management for COTS items.

NUREG/CR-6101, Section 3.1.3 "Software Configuration Management Plan," and Section 4.1.3, "Software Configuration Management Plan," contain guidance on SCMPs.

The software configuration management (CM) plan should exhibit the management, implementation and resource characteristics listed below.

B.3.1.11.1 Management Characteristics of the SCMP

The management characteristics that the SCMP should exhibit include purpose, organization, responsibilities, and security.

Purpose consists of a definition of the purpose and scope of the software CM activities. The SCMP should list the general functions the software CM organization will be expected to perform.

Organization involves a description of the software CM organization. The SCMP should describe the boundaries and interfaces between the CM organization and other company organizations. Reporting channels should be described.

Responsibilities involve a definition of the responsibilities and authority of the software CM organization. The SCMP should specify the person or group responsible for the successful completion of each CM task. It should define the duties of the configuration control board. It should specify the person who has the authority to release any software, data or documents for revision, and the person who has the authority to release any software, data or documents for operation after revision has been completed.

Security concerns configuration control measures to support SDOE per RG 1.152.

B.3.1.11.2 Implementation Characteristics of the SCMP

The implementation characteristics that the SCMP should exhibit include measurement, procedures, and record keeping.

Measurement concerns a set of indicators used to determine the success or failure of the software CM activity. The SCMP should require that data associated with configuration management be systematically collected and analyzed to determine the effectiveness of the CM effort. The SCMP should specify the criteria to be used to verify the completion of each CM task.

Procedures involve a description of the software configuration management strategy. The SCMP should specify procedures for identifying and naming configuration items. It should specify procedures for placing items under configuration control. It should describe the method for keeping data files and tables synchronized with the software that uses them, and for keeping software and its associated documentation synchronized. It should specify the procedure for associating source code with the derived object code and executable modules. Procedures should exist for managing software libraries. The SCMP should ensure the control and retrieval of qualification information associated with the software designs and code, software confirmation audits, and status accounting.

Items to be controlled should include: software requirements, designs, and code; support software used in development (exact versions); libraries of software components essential to safety; software plans that could affect quality; test software requirements, designs, or code used in testing; test results used to qualify software; analyses and results used to qualify software; software documentation; databases and software configuration data; pre-developed software items that are safety system software; software change documentation; and tools used in the software project for management, development or assurance tasks.

The SCMP should specify procedures for tracking problem reports, and for ensuring that each problem reported has been correctly resolved. The SCMP should describe the information required to approve a change request, and should ensure control of all software design changes. The relationship of software CM to other change control procedures, such as V&V anomaly handling and maintenance, should be described.

The SCMP should require periodic reviews and audits of the configuration baseline, including physical audits of the baseline.

The SCMP should include a description of the process used to maintain and track purchased items, such as software tools used to make the final product. A qualification procedure should be provided, and a method of tracking tool history, bug lists, and errata sheets should enable the applicant/licensee to track which design outputs may be affected by discovered tool or purchased item deficiencies. The SCMP should describe procedures to control vendors supplying safety system software.

Record keeping concerns a description of the software CM record keeping requirements. The SCMP should identify required CM records. Record structures (such as an annotated table of contents) should be provided. Procedures should exist for protecting configuration items. The SCMP should describe how configuration items will be stored, handled, retained and shipped. A tracking system should exist for managing configuration items, so that the revision history of each configuration item may be retrieved, and so that the latest revision of each configuration item may be easily identified. Procedures should exist for backup and disaster recovery.

B.3.1.11.3 Resource Characteristics for the SCMP

The resource characteristics that the SCMP should exhibit include methods/tools and standards.

Methods/tools refer to a description of the means that will be used to carry out each CM task. The SCMP should identify suitable facilities, methods, techniques and tools to facilitate the performance of the CM work. The SCMP should specify a process for selecting configuration management tools. The hardware and software environment within which the CM tools are to be applied and any necessary controls should be described.

Standards involves a list of the international, national, industry and company standards and guidelines to be followed by the software CM organization.

B.3.1.11.4 Review Guidance for the SCMP

The SCMP is another important plan because software errors can occur when changes to software are made to the wrong version of the software, or the changes are not sufficiently

tested to ensure that they do not introduce new errors. Configuration management starts once the initial software is initially released by the software design group.

One of the critical items to look for is an exact definition of who will control the software. There should be a software librarian or group who is responsible for keeping the various versions of the software, giving out the current version for test or modification, and receiving back the modified and tested software. Another critical item is that all software, not just the operational code to be used in the safety application, is controlled. This would include any software or software information which affects the safety software, such as software requirements, designs, and code; support software used in development; libraries of software components essential to safety; software plans that could affect quality; test software requirements, designs, or code used in testing; test results used to qualify software; analyses and results used to qualify software; software documentation; databases and software configuration data; pre-developed software items that are safety system software; software change documentation; and tools used in the software project for management, development or assurance tasks. Each of these can affect the final product if a wrong version is used during the software development process.
The SCMP should specify how modified software or documentation should be tested and verified, and who is to do this.

The SCMP may be two different plans, one used by the software vendor during the development of the software, and one used by the applicant/licensee during the operational phase of the project. The applicant/licensee plan may be contained in an overall plant configuration management plan. If this is the case, the reviewer should check that software specific issues have been addressed in the plant configuration management plan.

B.3.1.12 Software Test Plan (STP)

A STP should plan the implementation of all of the necessary testing activities. Guidance for the appropriate testing activities is found in the following sources.

RG 1.170, “Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” endorses IEEE Std 829, “IEEE Standard for Software Test Documentation,” with a few noted exceptions, identifies an acceptable method for satisfying test documentation requirements.

RG 1.171, “Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” which endorses ANSI/IEEE Std 1008, “IEEE Standard for Software Unit Testing,” with a few noted exceptions, identifies an acceptable method for satisfying software unit testing requirements.

The STP should exhibit the management, implementation and resource characteristics listed below.

B.3.1.12.1 Management Characteristics of the STP

The management characteristics that the STP should exhibit include purpose, organization, responsibilities, and security.

Purpose refers to a definition of the purpose and scope of the software testing activities. The STP should list the general functions the software testing organizations will be expected to perform.

Organization involves a description of the software testing organization(s). The STP should describe the boundaries and interfaces between the testing organization(s) and other company organizations. Reporting channels should be described.

Responsibilities involve a definition of the responsibilities and authority of the software testing organization(s). The STP should specify the person or group responsible for the successful completion of each testing task.

Security means that SDOE design features should be tested per RG 1.152.

B.3.1.12.2 Implementation Characteristics of the STP

The implementation characteristics that the STP should exhibit include measurement, procedures, and record keeping.

Measurement concerns a set of indicators used to determine the success or failure of the software testing activity. The STP should require that data associated with testing be systematically collected and analyzed to determine the effectiveness of the testing effort. The STP should specify the criteria to be used to verify the completion of each testing task.

Procedures involve a description of the software testing strategy. The STP should specify procedures for testing items. It should describe the method for keeping test cases and procedures synchronized with the software design. It should specify the procedure for associating released code with the associated testing documentation.

The STP should specify procedures for tracking problem reports, and for ensuring that each problem reported has been correctly resolved.

Record keeping concerns a description of the software testing record keeping requirements. The STP should identify required testing records. Procedures should exist for handling testing items. A tracking system should exist for managing test cases and procedure, so that the revision history of each item may be retrieved, and so that the latest revision of each item may be easily identified.

B.3.1.12.3 Resource Characteristics for the STP

The resource characteristics that the STP should exhibit include methods/tools and standards.

Methods/tools involve a description of the means that will be used to carry out each testing task. The STP should identify suitable facilities, methods, techniques and tools to facilitate the performance of testing. The STP should specify a process for selecting testing tools. The hardware and software environment within which the tests are to be applied and any necessary controls should be described.

Standards refers to a list of the international, national, industry and company standards and guidelines to be followed by the software testing organization.

B.3.1.12.4 Review Guidance for the STP

The STP should cover all testing done to the software, including unit testing, integration testing, factory acceptance testing, site acceptance testing and installation testing. The reviewer should examine the STP to ensure the test planning is understandable, that testing responsibilities have been given to the appropriate personnel, and that adequate provisions are made for retest in the event of failure of the original test. Since modifying software after an error occurs can result in a new error, it is important that the STP require the full set of tests be run after any modification to the software. The reviewer should ensure that since final system testing is considered a V&V test, the STP assigns the responsibility of the definition, test design, and performance to the V&V group.

B.3.2 Acceptance Criteria for Implementation

This subsection addresses acceptance criteria for implementation activities. The acceptance criteria address specific software life cycle process implementation activities and documentation. These activities and products, when found to be acceptable, provide the reviewer with confidence that the plans have been carried out.

The NRC staff reviewer confirms that the plans have been followed by the software developer. The detailed acceptance criteria are provided by the software developer and evaluated by the NRC staff in its acceptance of the plans. In addition to verifying that plans have been followed, the reviewer should pay particular attention to the areas discussed below. These activities are depicted in Figure 7-A-1 of this BTP as process implementation.

B.3.2.1 Acceptance Criteria for Safety Analysis Activities

The SSP describes the safety analysis implementation tasks that are to be performed. The acceptance criterion for software safety analysis implementation is that the tasks in that plan have been carried out in their entirety. Documentation exist show that the safety analysis activities have been successfully accomplished for each life cycle activity group and that the proposed digital system software is, in fact, safe. In particular, the documentation should show that the system safety requirements have been adequately addressed for each activity group; that no new hazards have been introduced; that the software requirements, design elements, and code elements that can affect safety have been identified; and that all other software requirements, design, and code elements will not adversely affect safety.

The safety analysis activities also should include the SDOE risk, and assessing the measures used to ensure that the design products do not contain undocumented code, and other unwanted or undocumented functions or applications.

B.3.2.2 Acceptance Criteria for Software Verification and Validation Activities

RG 1.168 endorses IEEE Std 1012, “IEEE Standard for Software Versification and Validation,” as providing methods acceptable to the staff for meeting the regulatory requirements as they apply to V&V of safety system software, subject to the exceptions listed.

RG 1.168 also endorses IEEE Std 1028, “IEEE Standard for Software Reviews and Audits,” as providing an approach acceptable to the staff for carrying out software reviews, inspections, walk through and audits, subject to the exceptions listed.

Clause 5.3.3, “Verification and Validation,” and Clause 5.3.4, “Independent Verification and Validation (IV&V) requirements,” of IEEE Std 7-4.3.2 which is endorsed by RG 1.152, contain guidance on V&V.

The SVVP describes the V&V implementation tasks that are to be carried out. The acceptance criterion for software V&V implementation is that the tasks in the SVVP have been carried out in their entirety. Documentation should exist that shows that the V&V tasks have been successfully accomplished for each life cycle activity group. In particular, the documentation should show that the requirements, design, code, integration, and installation design outputs satisfy the appropriate software development functional and process characteristics (as described in Subsection B.3.3 below).

Problems identified by the verification effort should be documented, together with any action items required to mitigate or eliminate each problem. A record should be kept of actions taken in response to the action items and the appropriate CM activities should be performed.

As part of the software V&V effort, a traceability analysis should be performed and documented. This traceability analysis documentation should clearly show the linkage between each requirement imposed on the software by the system requirements document and system design documents, and one or more requirements in the SRS. The analysis documentation should allow traceability in both directions. It should be organized so that as design, implementation, and validation take place, traceability information can be added for these activities. It should be updated at the completion of each life cycle activity group. The final analysis documentation should permit tracing from the system requirements and design through the software requirements, design, implementation, integration, validation, and installation.

The integration V&V activities should demonstrate that all unit and subsystem tests required by the SVVP were successfully completed. Any anomalies or errors found during the tests should be resolved and documented. Final integration tests should be completed and documented. Reports should be written for each test run. These reports should include any anomalies found and actions recommended. The final integration V&V report should describe the procedures followed and the tests performed during integration. This report should be consistent with the SIntP.

The software validation activities should demonstrate that all validation tests required by the SVVP were successfully completed. The testing process should contain one or more tests for each requirement in the SRS, as well as the acceptance criteria for each test. The result of each test should clearly show that the associated requirement has been met. Each test procedure should contain detailed information for the test setup, input data requirements, output data expectations, and completion time. Documentation should be produced for each test. Procedures should be included for handling errors and anomalies that are encountered during the testing. These procedures should include correction procedures (including configuration management), and provision for re-test until such time as the problems are resolved. A final report summarizing the validation testing should be provided. The report should contain a summary of problems and errors encountered during testing, and the actions taken to correct the problems encountered. The report should contain a statement that the validation testing was successful and that the software tested met all of the requirements of the SRS.

The installation (acceptance) test activities should document the test configuration, the required inputs, expected outputs, the steps necessary to execute the test, and the acceptance criteria for each test. The acceptance test activities should extend into post-installation testing performed during initial startup and ascension to full power. The procedure should require that problems

identified during the test activity, and any action items required to mitigate or eliminate each problem, be documented. Installation problems and their resolution should be documented. An acceptance test report should be produced describing the execution of the plan and summarizing the results. This report should contain a statement that the plan was successfully executed, and the system is ready for operation. The acceptance test report should demonstrate that the system operates correctly and is identical to the system that was validated during the validation phase. The report should summarize the test results after all problems have been satisfactorily resolved. The report should demonstrate that acceptance testing was executed according to the acceptance test procedure.

One of the accepted methods of checking the V&V effort is to perform a “thread audit.” This consists of picking a sample of plant parameters and tracing the software implementation of these parameters from the purchase specification and development of the functional requirements to the writing and testing of the code. The sample size should be sufficiently large to ensure a representative sample of the requirements and of the software code. This may be as many as 5% of the requirements. The minimum sample size should be determined by statistical significance criteria. This review includes:

(1) Reviewing actual sections of the code on a sample basis. Since the reviewer is seldom an expert in a particular language, this may necessitate that the responsible software design engineer walk the reviewer through the code. If the reviewer is unable to follow this explanation, this portion of the thread audit should be delegated to a more experienced staff person or an independent contractor.

(2) Examining the various levels of software development documents and comparing them to the code.

(3) Examining problem reports and test plans for the selected requirements, and verifying the corrections. It would be unusual if there were no problem reports, and if this is the case, the testing and review procedures should be carefully examined to ensure that a thorough test and review was done. Each of the completed problem reports should show what was done to resolve the problem, and how that resolution was tested. This would also be a good time to check the configuration management procedures to see how the revised code was put under configuration management.

(4) Examining the engineering cross-discipline interfaces to ensure that nuclear specific needs were correctly incorporated into the code.

(5) Examining the applicant or licensee interface to ensure plant specific requirements are correctly incorporated.

(6) Ensuring that the V&V process is followed according to the vendor's plan.

(7) Reviewing the final results of the process.

There may be other items in the thread audit, depending on the software tools used and the exact nature of the programing requirements and the methods. The reviewer should not hesitate to ask why something was done in a particular manner, and to use experience and judgment to assess the answer.

If errors are found, the appropriate V&V records should be examined to see if the V&V team has also caught the errors. If the requirements, code and test have been verified and validated without finding the error, a serious quality problem may exist. Additional requirements should then be checked to see if this is a systematic or an isolated problem. If several of these problems exist, the adequacy of the software development process may be insufficient to produce high quality software for use in safety-related applications in nuclear power plants. The reviewer should discuss these concerns with management to determine if the applicant or licensee request to use this software should be rejected.

A successful thread audit necessitates that the reviewer be familiar with the various plans and procedures used for the software development and life cycle, and the various NRC requirements, NUREGs, RGs, and industry standards that those plans and procedures are based upon. This knowledge is necessary to be able to determine if the methodologies examined during the thread audit are those which the software developer committed to using, and if they are being used correctly. The thread audit is one of the few times where the reviewer can examine the actual development process rather than the documented plans for a development process.

B.3.2.3 Acceptance Criteria for Software Configuration Management Activities

RG 1.169, "Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants," endorses IEEE Std 828, "IEEE Standard for Software Configuration Management Plans," subject to specific provisions identified in the RG, as providing guidance that is acceptable for carrying out software configuration management.

Software configuration management can be partitioned into two types of activities: the management and control of the software and the associated development environment; document control. There are many configuration management tools available for software. A particular tool should be selected, evaluated, and used properly for software configuration control.

The SDP describes the software and documents that will be created and placed under configuration control. If the software and documents are controlled in different systems, then each should be described, as well as their relationship to each other. The SCMP describes the implementation tasks that are to be carried out. The acceptance criterion for software CM implementation is that the tasks in the SCMP have been carried out in their entirety. Documentation should exist that shows that the configuration management tasks for that activity group have been successfully accomplished. In particular, the documentation should show that configuration items have been appropriately identified; that configuration baselines have been established for the activity group; that an adequate change control process has been used for changes to the product baseline; and that appropriate configuration audits have been held for the configuration items created or modified for the activity group.

Each configuration item should be labeled unambiguously so that a basis can be established for the control and reference of the configuration items defined in the SCMP. Configuration baselines should be established for each life cycle activity group, to define the basis for further development, allow control of configuration items, and permit traceability between configuration items. The baseline should be established before the set of activities can be considered complete. Once a baseline is established, it should be protected from change. Change control activities should be followed whenever a derivative baseline is developed from an established

baseline. A baseline should be traceable to the baseline from which it was established, and to the design outputs it identified or to the activity with which it is associated.

Configuration control actions should be used to control and document changes to configuration baselines. A configuration control board (CCB) should exist with the authority to authorize all changes to baselines. Problem reports should be prepared to describe anomalous and inconsistent software and documentation. Problem reports that require corrective action should invoke the change control activity. Change control should preserve the integrity of configuration items and baselines by providing protection against their change. Any change to a configuration item should cause a change to its configuration identification. This can be done via a version number or attached change date. Changes to baselines and to configuration items under change control should be recorded approved and tracked. If the change is due to a problem report, traceability should exist between the problem report and the change. Software changes should be traced to their point of origin, and the software processes affected by the change should be repeated from the point of change to the point of discovery. Proposed changes should be reviewed by the CCB for their impact on system safety.

Status accounting should take place for each set of life cycle activities prior to the completion of those activities. The status accounting should document configuration item identifications, baselines, problem report status, change history and release status.

The configuration management organization should audit life cycle activities to confirm that configuration management procedures were carried out in the life cycle process implementation.

B.3.2.4 Acceptance Criteria for Testing Activities

RG 1.152, “Criteria for Use of Computers in Safety Systems of Nuclear Power Plants,” which endorses IEEE Std 7-4.3.2, “IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations,” with a few noted exceptions, identifies an acceptable methods for satisfying computer system qualification testing requirements (See: IEEE Std 7-4.3.2 Clause 5.4.1, “Computer System Testing”).

RG 1.168, Section C.7.b, “Regression Analysis and Testing,” and C.7.d, “Test Evaluation,” contain guidance related to testing activities.

RG 1.170, “Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” endorses IEEE Std 829, “IEEE Standard for Software Test Documentation,” with a few noted exceptions, identifies an acceptable method for satisfying test documentation requirements.

RG 1.171, “Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” which endorses ANSI/IEEE Std 1008, “IEEE Standard for Software Unit Testing,” with a few noted exceptions, identifies an acceptable method for satisfying software unit testing requirements.

Thorough software testing consists of testing the smallest testable units, and then integrating those units into larger testable units, and testing that integrated unit. This process is repeated until finally the system is tested after installation. There is no generally accepted, precise definition of a unit, module, and system. It is therefore understood that RG 1.171 applies to all testing before integrated system testing.

The software development process characteristics listed below should be exhibited by testing activities.

Completeness means that all required functionality for all operating modes, including error recovery be tested.

Consistency means that the testing is consistent with the safety system requirements, the safety system design, the SRSs, the SDS, and documented descriptions and known properties of the operational environment within which the safety system will operate. Uniform and consistent terminology, notation, and definitions should be used throughout.

Traceability means that a trace exists between the SRS, the SDS, and the test documentation, which shows how each requirement is tested.

B.3.3 Acceptance Criteria for Design Outputs

This subsection describes the criteria to be used to determine whether the software has each of the characteristics important to safety system software. Criteria are organized first by life cycle activity group, then by design output, and then by characteristic.

Formal or semiformal methods are available for use in preparing some of the design outputs described in this subsection. Subsection C.3 of SRP Appendix 7.0-A describes the benefits of using such methods and the precautions that should be observed when reviewing design outputs prepared with such methods.

Acceptance criteria are divided into two sets: functional characteristics and process characteristics, as shown in the following table. Not all characteristics occur for every design output.

Functional Characteristics	**Process Characteristics**
Accuracy	Completeness
Functionality	Consistency
Reliability	Correctness
Robustness	Style
Safety	Traceability
Security	Unambiguity
Timing	Verifiability

Functional Characteristics

Certain system level functional criteria are required by 10 CFR 50.55a(h), and these shall be supported by the software.
The following paragraphs define the acceptance criteria for the functional characteristics for safety system software.

Accuracy - These accuracy values should also be used in setpoint determination calculations.

Functionality - The functionality of the proposed system shall be compared to the plant requirements. The overall system, hardware and software architecture needs to be evaluated to ensure that the functionality requirements can be met.

Reliability - The reliability of software based digital systems is difficult to determine, and to date, the state-of-the-art is such that no consensus exists on software reliability prediction methods. Care must be taken when crediting any reliability values.

Robustness - The robustness of a software system needs to be specifically tested before any robustness characteristics can be accepted. The tests used should be specifically analyzed to determine if the tests do indeed test for the robustness characteristic, such as invalid inputs or environmental stressors, are actually being validated.

Safety - The safety of the digital system should be explicitly considered in the design process, and documented in the design outputs.

Security - The establishment of a SDOE for digital safety systems per RG 1.152.

Timing - The system timing should be analyzed for indication of deterministic behavior.

Process Characteristics

Completeness - This characteristic is difficult to define and check, and depends to a large degree on the thoroughness of the review and the experience of the reviewer. A requirements compliance matrix, showing all system requirements and where in hardware and software, software code, test and the V&V process each of these individual requirements was address is valuable. The method the applicant or licensee used to reach the determination of completeness should be reviewed.

Consistency - Determination of consistency requires careful examination of the individual hardware and software components and a thorough understanding of the inputs and outputs of each. For purchased items which undergo commercial grade dedication, insight into the design process used for these purchased items may be required. Each plan should be internally consistent, and the complete set of plans should be mutually consistent.

Correctness - One of the often overlooked aspects of correctness is simplicity. The system and specifically the software should be no more complex than required to perform the needed functions. Unused or unneeded functions and code should not be in the safety-related software, even if the software developer wishes to include them for ease of use, future development, or other reasons. The system and software requirements and the final code should be examined to insure that only those features need to implement the safety functions and to perform system and software testing are included.

Style - The style of the planning documents is generally determined by the requirement for those documents, such as IEEE standards or individual applicant or licensee requirements. The style of design output documentation is generally determined by the planning documents. It is generally more important that the styles be consistent within a project and be understandable than that the style be of any particular type. Programing style will be dependent on the coding standards selected for use. The coding standards should be examined for understandability and the resulting code should meet the coding standard. Various software language requirements and practices are discussed in NUREG/CR-6463, “Review Guidelines for Software Languages for Use in Nuclear Power Plant Safety Systems.”

Traceability - A Requirements Traceability Matrix, which shows every requirement, broken down in to sub-requirements as necessary, and what portion of the software requirement, software

design description, actual code, and test requirement addresses that system requirement. This is central to for the production of complex systems to ensure all requirements are implemented, checked and tested.

Unambiguity - This is one of the most difficult characteristics to review. The software development community and the nuclear community may use the same words in different manners, and therefore each group may read statement, and each may believe it is unambiguous, but each may have a different understanding of the meaning, and therefore a different understanding of what is required. Experience of the staff reviewer is important, and any element or statement which may have more than one meaning should be discussed with both the vendor and applicant or licensee personnel to ensure both have the same understanding. If the element is not clear to the reviewer, it is likely that others may also share that lack of understanding.

Verifiability - The degree to which a software planning document, implementation process document or design output is stated or provided in such a way as to facilitate the establishment of verification criteria and the performance of analyses, reviews, or tests to determine whether those criteria have been met. A good example of a verifiable statement is a statement of conformance to a standard. Each requirement in the standard can then be checked to verify conformance (assuming that the requirements within the standard are verifiable). Partial conformance statements, like “meets the intent of” or “consistent with” are unverifiable unless there is an accompanying identification of which requirements, recommendations, and permissions are being followed.

B.3.3.1 Requirements Activities - Software Requirements Specification (SRS)

RG 1.172, “Software Requirements Specifications for Digital Computer Software Used in Safety Systems of Nuclear Power Plants,” endorses IEEE Std 830, “IEEE Recommended Practice for Software Requirements Specifications,” with a few noted exceptions, describes an acceptable approach for preparing software requirements specifications for safety system software.

NUREG/CR-6101, Section 3.2.1, “Software Requirements Specification,” and Section 4.2.1, “Software Requirements Specifications,” contain relevant guidance.

An SRS that exhibits the functional and the software development process characteristics listed below should be produced.

B.3.3.1.1 Functional Characteristics of SRS

For each of the functional characteristics, the requirements imposed by the system requirements and system design on the software for that characteristic should be contained in the SRS. Functional characteristics addressed by the SRS include accuracy, functionality, reliability, robustness, safety, security, and timing.

Accuracy means that accuracy requirements should be provided for each input and each output variable. Accuracy requirements should be stated numerically, and appropriate physical units and error bounds should be supplied. Accuracy requirements should include a description of data type and data size for each input and output variable.

Functionality means that the operations that must be performed for each mode of operation are completely specified. Functions should be specified in terms of inputs to the function, transformations to be carried out by the function, and outputs generated by the function.

Reliability means that all requirements for fault tolerance and failure modes are fully specified for each operating mode. 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 provided.

Robustness means that the behavior of the software in the presence of unexpected, incorrect, anomalous and improper: (1) input, (2) hardware behavior, or (3) software behavior is fully specified. Of particular concern is the behavior of the software in the presence of unexpectedly high or low rates of message traffic.

Safety means that the software functions, operating procedures, input, and output be classified according to their importance to safety. Requirements important to safety should be identified as such in the SRS. The identification of safety items should include safety analysis report requirements, as well as abnormal conditions and events as described in RG 1.152.

Security means the establishment of SDOE per RG 1.152.

Timing means that functions that must operate within specific timing constraints are identified, and that timing criteria are specified for each. Timing criteria should be provided for each mode of operation. Timing requirements should distinguish between goals and requirements. Timing requirements should be stated in such a way that the time delay between stimulus and response for safety actions is deterministic under normal and anticipated failure conditions. SRP BTP 7-21 provides additional guidance on real-time performance.

B.3.3.1.2 Process Characteristics of SRS

Software development process characteristics exhibited by the SRS should include completeness, consistency, correctness, style, traceability, unambiguity and verifiability.

Completeness means that all actions required of the computer system are fully described for all operating modes and all possible values of input variables (for example, the complete span of instrument inputs or clock/calendar time).[*] The SRS should describe any actions that the software is prohibited from executing. The operational environment within which the software will operate should be described. All variables in the physical environment that the software must monitor and control should be fully specified. Functional requirements should describe: how each function is initiated; (2) the input and output variables required of the function; (3) the task sequences, actions, and events required to carry out the function; and (4) the termination conditions and system status at the conclusion of the function. User interfaces should be fully described for each category of user.

Consistency means that the contents of the SRS are consistent with the safety system requirements, the safety system design, and documented descriptions and known properties of the operational environment within which the safety system software will operate. Individual

[*] Implementation of safety functions should not rely upon a date (calendar time). Actions depending upon calendar time should account for concerns such as those identified with the year 2000 (two-digit 00).

requirements should not contradict other requirements. Timing requirements should be consistent with thermo-hydraulic analyses performed in the system safety analysis. Uniform and consistent terminology, notation, and definitions should be used throughout the SRS.

Correctness means that the description of actions required of the computer system are free from faults and that no other requirements are stated. The operational environment within which the software will operate should be accurately described. All variables in the physical environment that the software must monitor and control should be properly specified. Functional requirements should accurately describe: (1) how each function is initiated; (2) the input and output variables required of the function; (3) the task sequences, actions and events required to carry out the function; and (4) the termination conditions and system status at the conclusion of the function.

Style means that the contents of the SRS are understandable. The SRS should differentiate between requirements placed on the software and other supplementary information, such as design constraints, hardware platforms, and coding standards. A precise definition of each technical term should exist, either in the SRS or in a separate dictionary or glossary. Each requirement should be uniquely and completely defined in a single location in the SRS.

Traceability means that a two-way trace exists between each requirement in the SRS and the safety system requirements and design. There should be a two-way trace between each requirement in the SRS and the software design, as well as a forward trace from each requirement in the SRS to the specific inspections, analyses, or tests used to confirm that the requirement has been met.

Unambiguity means that each requirement, and all requirements taken together, have one and only one interpretation.

Verifiability means that it is possible to construct a specific analysis, review, or test to determine whether each requirement has been met.

B.3.3.1.3 Review Guidance for SRSs

Errors in requirements or misunderstanding of requirement intent are a major source of software errors. The requirements should be carefully examined by the reviewer, and each of the above functional characteristics should be present in each requirement. If the requirements are not clear to the reviewer, they will probably not be clear to the software design team.

The thread audit discussed above is a tool which can be used to check each of these characteristics. During the thread audit, for each requirement traced, the reviewer should check that each requirement is complete, that the requirements are consistent with the overall safety system requirements, and that the requirement is not in conflict with some other requirement. The requirements should be understandable and unambiguous. Each requirement should be traceable to one or more safety system requirements, and the requirements traceability matrix should show where in the software the required action is being performed. The requirements traceability matrix should also show where the particular requirement is being tested.

B.3.3.2 Design Activities - Software Architecture Description (SAD)

NUREG/CR-6101, Section 3.3.1, “Hardware and Software Architecture,” and Section 4.3.1, “Hardware/Software Architecture Specifications,” contain relevant guidance.

A SAD should be produced. The SAD should include all of the functional and software development process characteristics listed below.

B.3.3.2.1 Functional Characteristics of SAD

For each of the functional characteristics, the requirements imposed on the software for that characteristic should be satisfied by the software architecture. A review of the software architecture requires a concurrent review of the hardware architecture. Functional characteristics addressed by the SAD should include reliability, safety, security, and timing.

Reliability means that the combined hardware and software architecture are such that individual software element failure will not compromise safety. The software architecture should identify actions to be taken in the event of error detection. The hardware and software architecture should be reviewed to verify that the propagation of errors is controlled via a well-structured modular design.

Safety means that the software architecture introduces no new hazards into the safety system. The safety functions should be separated from normal operating and overhead functions, with well-defined and strictly controlled interfaces between them. Any online maintenance features should be included. The hardware and software architecture should be reviewed to verify that there is no violation of other criteria such as single failure, channel separation, and separation between Class 1E and non-1E systems. The review should verify that no new hazards are introduced into the safety system as a result of the architecture configuration.

Security means the establishment of a SDOE per RG 1.152.

Timing means that the architectural design describes all timing limitations, the strategy for handling each, the required margins, and the method of measuring those margins. A timing specification should exist for each architectural element, in terms of minimum and maximum times for execution. Scheduling mechanisms and inter-process communication methods should be described. The architecture should be such that operations are performed in the correct sequence. SRP BTP 7-21 provides additional guidance on real-time performance. SRP Section 7.9 provides additional guidance on digital data communications systems.

B.3.3.2.2 Process Characteristics of SAD

The software development process characteristics that the SAD should exhibit include completeness, consistency, style, traceability, and verifiability.

Completeness means that all the software requirements are satisfied in the architecture. The SAD should address all operating modes specified in the SRS, including initialization, operational, shut-down, maintenance, and test modes.

Consistency means that each software architectural element is compatible with the SRS, the hardware architecture, documented descriptions and known properties of the operational and hardware environment, and other software elements. Timing specifications of each software element should be consistent with the specifications of the other elements with which it interacts and with the expected performance of the system as a whole. Uniform and consistent terminology, notation, and definitions should be used.

Style means that the contents of the SAD are understandable. The architecture description should conform to the developer's style guide. The architecture specification should contain the rationale for architectural decisions.

Traceability means that a two-way trace exists between the requirements in the SRS and the elements in the architecture. A two-way trace should exist between the architectural elements and the detailed design elements. There should be a forward trace from each architectural element to the specific inspections, analyses, or tests that will be used to confirm that the element has been correctly designed.

Verifiability means that it is possible to construct specific analyses, reviews, and tests to verify that the architecture satisfies the software requirements.

B.3.3.2.3 Review Guidance for SAD

The reviewer should be able to refer to this architecture to understand how the software works, the flow of data, and the deterministic nature of the software. The architecture should be sufficiently detailed to allow the reviewer to understand the operation of the software.

B.3.3.3 Design Activities - Software Design Specification (SDS)

NUREG/CR-6101, Section 3.3.2, “Software Design Specification,” and Section 4.3.2, “Software Design Specifications,” contain relevant guidance.

An SDS should be produced. The SDS should include all of the functional and software development process characteristics listed below.

B.3.3.3.1 Functional Characteristics of SDS

For each of the functional characteristics, the requirements imposed on the software for that characteristic should be satisfied by the software design. Product functional characteristics addressed by the SDS should include accuracy, reliability, robustness, safety, security, and timing.

Accuracy means that all calculations are specified in such a way that the accuracy requirements for the calculations will be satisfied. In particular, floating point arithmetic should be avoided; if that is not possible, special care must be taken to maintain the accuracy of the calculations. The design should specify the method for determining that the values of input variables are within the proper range, the method by which the software will detect that the values of input variables are not within their proper range, and the actions to be taken in the latter case. All calculations should be analyzed for convergence, round-off error, precision, and accuracy as appropriate.

Reliability means that the detailed software design is such that single failures of individual elements will not cause safety system failure.

Robustness means that the design is such that the software will operate correctly in the presence of unexpected, incorrect, anomalous and improper: (1) input, (2) hardware behavior, or (3) software behavior. In particular, the software should not fail, and should not provide incorrect outputs, in the presence of these conditions. Attention should be paid to those values

of input variables that are physically possible to the device, even if logically impossible in the application (to account for sensor errors, communication line noise, and similar concerns).

Safety means that the detailed design introduces no new safety hazards into the safety system.

Security means that unauthorized changes are prevented, detected, or mitigated as appropriate per RG 1.152.

Timing means that the time delay between stimulus and response is deterministic. SRP BTP 7-21 provides additional guidance on real-time performance.

B.3.3.3.2 Process Characteristics of SDS

The SDS should exhibit each of the following software development process characteristics: completeness, consistency, correctness, style, traceability, and verifiability.

Completeness means that the detailed design specifies the actions of each software unit for the entire domain of each input variable (for example, the complete span of instrument inputs or clock/calendar time). The design should be sufficiently complete to permit implementation to take place. Actions should be specified for all situations anticipated in the SRS. Equipment, human, hardware, and software interfaces should be correctly and fully specified. Equations, algorithms, and control logic should be correctly and fully specified.

Consistency means that the detailed design is consistent with the architectural design, and that the detailed design elements are mutually consistent. Design elements should be consistent with documented descriptions and known properties of the operational environment within which the software will execute. Input and output specifications specified in the software design should be consistent with interface requirements imposed by the hardware or pre-developed software products. Timing specifications of each detailed design element should be consistent with the timing specifications of the architectural element of which it is a part. Models, algorithms, and numerical techniques specified in the software design should agree with standard references where such are applicable. A uniform and consistent terminology, notation, and definitions should be used. Models, algorithms, and numerical techniques specified in the software design should be mathematically mutually compatible.

Correctness means that all equations, algorithms, and control logic are evaluated for potential errors. All equations and algorithms should be defined to a sufficient level of detail to permit coding. Data structure design should ensure that the code elements will correctly initialize data, correctly access stored data, and correctly scale and dimension data. The detailed design should ensure that no data item can be used before it is initialized, can have its value changed in an unanticipated manner, or can have its value changed by an unanticipated design element. The detailed design should ensure that no data item can be changed in an unanticipated manner.

Style means that the detailed design documents description should conform to the developer's style guide. Each element of the detailed design should be specified. The detailed design documentation should contain the rationale for design decisions. Programming language standards should be identified. The detailed design documentation should identify those language features which will not be used without justification.

Traceability means that a two-way trace exists between the elements of the detailed design and the elements in the architecture. A two-way trace should exist between the detailed design elements and the code elements. There should be a forward trace from each detailed design element to the specific inspections, analyses, or tests that will be used to confirm that the element has been correctly designed.

Verifiability means that it is possible to construct specific analyses, reviews, and tests to verify that the design satisfies the software architecture.

B.3.3.3.3 Review Guidance for SDS

The SDS is primarily used to ensure that the software code accurately reflects the software requirements. The thread audit should check several of the requirements and follow them through the final code, but the entire SDS should be read by the reviewer to determine that it is understandable, and contains sufficient information. In addition, the V&V report on the SDS should be carefully reviewed. The reviewer should not be able to find any problems which have not been found and documented by the V&V team. If the SDS is reviewed after completion of the V&V effort, the reviewer should find no errors.

B.3.3.4 Implementation Activities - Code Listings (CL)

NUREG/CR-6463, Revision 1, "Review Guidelines for Software Languages for Use in Nuclear Power Plant Safety Systems," contains relevant guidance.

A software implementation (code) should be produced. The code should include all of the functional and process characteristics listed below.

B.3.3.4.1 Functional Characteristics of CL

For each of the functional characteristics, the requirements imposed on the software for that characteristic should be satisfied by the code. Functional characteristics addressed by the code documents should include accuracy, robustness, safety, and timing.

Accuracy means that the actual source code is written so that the accuracy requirements and accuracy design specifications are met. In particular, special care should be taken for floating point arithmetic, round-off errors, and the retention of precision during numerical operations. If mathematical subroutine libraries are used, the accuracy characteristics of the subroutines should be known and documented, and shown to meet the accuracy requirements and accuracy design specifications.

Robustness means that the system is coded in such a way that corrupted data will not cause the safety system to fail. Data corruption should be avoided. All input data should be checked to ensure that the correct data is being read and that the data is in the correct format. All messages should be checked to ensure that the correct message is being read and that the message contents are in the correct format. Appropriate corrective actions should take place if any of these criteria are violated.

Safety means that the code introduces no new hazards into the safety system.

Security means that unnecessary codes are not introduced into the safety system software per RG 1.152.

Timing means that the execution time is deterministic. SRP BTP 7-21 provides additional guidance on real-time performance.

B.3.3.4.2 Process Characteristics of CL

Software development process characteristics exhibited by the code documents should include completeness, consistency, correctness, style, traceability, and verifiability.

Completeness means that the code meets all the specifications of the design and all implementation constraints. The software implementation should be compatible with the hardware environment.

Consistency means that all variable names, types, locations, and array sizes are defined consistently throughout the software units. The code should use mathematical equations which correspond to the mathematical models, algorithms, and numerical techniques described in or derived from the SDS. All parameters passed between software units should be consistent with respect to number, type, structure, physical units, and direction. Minimum and maximum execution times should be consistent with expected overall performance.

Correctness means that the code implements all of the requirements in the SDS, and contains no features or functions that are not required.

Style means that the programming style constraints specified in the design documents are followed. NUREG/CR-6463, "Review Guidelines on Software Languages for Use in Nuclear Power Plant Safety Systems," provides guidance on coding practices to be avoided. In particular, data structures should be protected so that they cannot be changed simultaneously. Arrays should have a fixed, predefined length. Global variables and dynamic memory allocation should not be used.

Traceability means that a two-way trace exists between the elements of the detailed design and the elements in code. A two-way trace should exist between the code elements and the specific software subsystem or system which contains that element during factory build and test. There should be a forward trace from each detailed design element to the specific inspections, analyses, or tests that will be used to confirm that the element has been correctly implemented.

Verifiability means that it is possible to construct specific analyses, reviews, and tests to verify that the code correctly implements the detailed design.

B.3.3.4.3 Review Guidance for CL

In addition to the above, the CL should have sufficient comments and annotations that the intent of the code developer is clear. This is not only so the reviewer can understand and follow the code, but also so future modifications of the code are facilitated. Undocumented code should not be accepted as suitable for use in safety-related systems in nuclear power plants. The documentation should be sufficient for a qualified software engineer to understand. If the reviewer does not have enough experience in this particular language or with the software tool being used, the reviewer may require the assistance of other NRC personnel or independent contractor personnel to make this determination.

B.3.3.5 Integration Activities - System Build Documents (SBDs)

NUREG/CR-6101, Section 3.5.1, “System Build Documents,” and Section 4.5.1, “System Build Documents,” contain relevant guidance.

One or more system build documents should be produced. The build documents should include all of the functional and software development process characteristics listed below.

B.3.3.5.1 Functional Characteristics of SBDs

For each of the functional characteristics, the requirements imposed on the build documents for that characteristic should be satisfied. Functional characteristics addressed by the build documents should include robustness, safety, and security.

Robustness means that the software build documents specify methods to detect incorrectly built software releases. The software build documents should identify all errors and anomalies discovered during software build activities.

Safety means that the software build activity introduces no new hazards into the safety system.

Security means that the software build activity does not introduces unintended changes into the safety system software per RG 1.152.

B.3.3.5.2 Process Characteristics of SBDs

The system build documents should exhibit each of the following software development process characteristics: completeness, consistency, correctness, style, traceability, and verifiability.

Completeness means that all build procedures are fully specified. The software build documents should include all required software units, including code and data, that are part of the build.

Consistency means that the software build documents are consistent with the software specifications, as described in the SRS, software design description, and software code. A consistent and uniform set of terminology, notation, and definitions should be used throughout the software build document.

Correctness means that the software build documents identify the correct versions of all required software elements and all required software documents. It should be verified that the correct elements have actually been used in the build, including proper units from software libraries.

Style means that the software build documents conform to applicable standards imposed by the developer. A precise definition of each technical term used in the build documents should be included in the document, or in a separate dictionary or glossary.

Traceability means that it is possible to trace each element of the integrated builds (software subsystem or software system) backward to the code elements contained in the build. It should be possible to trace each element of the integrated build forward to the software field installation.

Verifiability means that it is possible to analyze, review, or test each integrated software build for the product functional requirements. The system build documents should specify methods to detect incorrectly built software releases. The build documents should identify all errors and anomalies discovered during software build activities.

B.3.3.5.3 Review Guidance for SBDs

The SBD is needed to verify that the software actually delivered and installed on the safety system is the software which underwent the V&V process and was tested. Any future software maintenance will depend on the maintainers knowing which version of the software to modify. The reviewer should check to ensure that the software listed in the build documentation is identified by version, revision, and date. The reviewer should also verify that this is the version and revision which was tested. This information should all be available from the configuration management group.

B.3.3.6 Installation Activities - Installation Configuration Tables (ICTs)

Installation configuration tables should be produced. They should include all of the functional characteristics listed below to ensure that the software will be correctly configured in the operating safety system. The software development process characteristics listed below should be exhibited by the installation configuration tables themselves.

B.3.3.6.1 Functional Characteristics of ICTs

For each of the functional characteristics, the requirements imposed on the configuration tables for that characteristic should be satisfied. Functional characteristics addressed by the configuration tables should include functionality, safety, and security.

Functionality means that the installation tables configure the installed system to have the functionality that is required for the plant.

Safety means that the installation tables introduce no new hazards into the safety system.

Security means that the installation tables do not introduce unneeded functionality into the installed system, and that the installation tables are protected from unauthorized changes per RG 1.152.

B.3.3.6.2 Process Characteristics of ICTs

The configuration tables should exhibit each of the specified software development process characteristics: completeness, consistency, correctness, traceability, and verifiability.

Completeness means that the software configuration tables include all information necessary for the correct operation of the system.

Consistency means that the installation configuration tables are consistent with the software specifications, as described in the SRS, software design description, software code, and software build documents.

Correctness means that the software configuration tables contain all plant-specific data.

Traceability means that it is possible to trace each installed program element backward to the integrated software elements that created that installed program element.
Verifiability means that it is possible to analyze, review, or test each installed software system on initial software installation, all subsequent installations, and periodically during operation.

B.3.3.6.3 Review Guidance for ICTs

In the event that the software has options for use, variable setpoints or other data, or may operate in various methods, the software needs to be configured for the particular plant requirements. Any software item which is changeable should have the intended configuration recorded in the ICT, and the reviewer should sample these configuration items to verify that they are correct. The reviewer should verify that the V&V team has already made this determination, and should then sample various items. For example, the reviewer could verify that setpoints shown in the ICT agree with values determined by setpoint calculations. Other configuration items may require reference back to the original system specification. The ICT is a critical item for configuration management, both by the vendor and by the applicant or licensee.

B.3.3.7 Installation Activities - Operations Manuals (OMs)

NUREG/CR-6101, Section 3.7.1, "Operations Manual," contains relevant guidance.

One or more software OMs should be produced. They may be incorporated into a system operations manual. Because operations manuals do not impose requirements on the software itself, the functional characteristics described in other sections of this BTP are not directly relevant. However, the software development process characteristics listed below should be exhibited by software operations manuals.

B.3.3.7.1 Process Characteristics of OMs

The OMs exhibit each of the specified software development process characteristics: completeness, consistency, style, traceability, and unambiguity.

Completeness means that all actions available to the system operator are fully described for all operating modes, including error recovery and backup. Operator actions should be specified in terms of inputs supplied by the operator or equipment, actions initiated by the operation, and responses to the operator. The purpose and operation of each function should be described, including interfaces with other functions. The operations manual should describe the operational environment within which the software will operate, including precautions and limitations that must be observed during operations to avoid exposing personnel or the plant to hazards or security vulnerabilities. All variables in the physical environment that the software must monitor and control should be fully described. User interfaces should be fully described for each category of user.

Consistency means that the operations manual is consistent with the system operations, safety system requirements, the safety system design, the SRSs, the SDS, and documented descriptions and known properties of the operational environment within which the safety system will operate. Individual user instructions should not contradict other instructions. Uniform and consistent terminology, notation, and definitions should be used throughout the operations manuals.

Style means that the operations manual is understandable by the users of the manual. A precise definition of each technical term should exist, either in the operations manual or in a separate dictionary or glossary. The operations manual may be organized in the style of a reference manual, with the assumption that its users are well trained.

Traceability means that a forward trace exists between the SRS, the operations plan, and the operations manual, which shows how each requirement is to be carried out by the operators, or carried out automatically by the safety system without operator action, and how the results of each requirement are displayed to the operators. A forward trace should also exist from all error messages generated by the code to a description of the error messages in the operations manual.

Unambiguity means that instructions to users have only one interpretation for the users.

B.3.3.7.2 Review Guidance for OMs

The reviewer should keep in mind that the intent of the staff review of manuals is to ensure that digital system is safe, and therefore the portion of the operations manuals which is of primary concern is the portion which deals with operation of the system under unusual or emergency conditions. The portion of the manual which deals with normal operation should be reviewed, but does not require the depth of review that emergency operations does.

B.3.3.8 Installation Activities - Software Maintenance Manuals (SMMs)

NUREG/CR-6101, Section 3.7.4, “Maintenance Manuals,” contains relevant guidance.

One or more SMM should be produced. They may be incorporated into a system maintenance manual. Because maintenance manuals do not impose requirements on the software itself, the functional characteristics described in other subsections of this BTP are not directly relevant. However, the software development process characteristics listed below should be possessed by software maintenance manuals.

B.3.3.8.1 Process Characteristics of SMMs

The SMMS should exhibit each of the specified software development process characteristics: completeness, style, and traceability.

Completeness means that maintenance procedures are fully defined. This should include identification of precautions and limitations that must be observed during maintenance to avoid exposing personnel or the plant to hazards or security vulnerabilities. Trouble reports should be collected from field installations and analyzed to determine if changes to the software are needed. Configuration management procedures should be described in or referenced by the maintenance manual. Procedures should exist to: (1) verify that changes have been carried out correctly and that no faults have been introduced in the software by the changes, and (2) ensure that software is correctly returned to service. Field upgrade procedures should be described.

Style means that the maintenance manual is understandable to the users. A precise definition of each technical term should exist, either in the maintenance manual or in a separate dictionary or glossary. The maintenance manual may be organized in the style of a reference manual, with the assumption that its users are well trained.

Traceability means that a forward trace exists between the maintenance plan and the maintenance manual, which shows how each requirement is carried out by the maintenance organization.

B.3.3.8.2 Review Guidance for SMMs

Prior to the review of the maintenance manuals, the reviewer should determine if maintenance will be done by applicant or licensee or vendor personnel. In many instances, the applicant or licensee only performs maintenance to replace failed circuit boards, and the boards are then sent to the vendor for repair. In this instance, the maintenance manuals used by applicant or licensee personnel do not need to be as detailed as would be required if the applicant or licensee was doing board level repairs. The reviewer should use judgment to determine how adequate the level of detail is in each of the maintenance manuals.

B.3.3.9 Installation Activities - Software Training Manuals (STMs)

NUREG/CR-6101, Section 3.7.3, “Training Manuals,” contains relevant guidance.

One or more STMs should be produced. They may be incorporated into a system training manual. Because training manuals do not impose requirements on the software itself, the functional characteristics described in other subsections of this BTP are not directly relevant. However, the software development process characteristics listed below should be exhibited by software training manuals.

B.3.3.9.1 Process Characteristics of STMs

The STMS should exhibit each of the specified software development process characteristics: completeness, consistency, style, and traceability.

Completeness means that all actions available to the operator are fully described for all operating modes, including error recovery. Operator actions should be specified in terms of inputs supplied by users and equipment, actions initiated by the operation, and responses to the user. The training manual should describe the operational environment within which the software will operate, including precautions and limitations that must be observed during operations to avoid exposing personnel or the plant to hazards. All variables in the physical environment that the software must monitor and control should be fully described. User interfaces should be fully described for each category of user.

Consistency means that the training manual is consistent with the safety system requirements, the safety system design, the SRSs, the SDS, and documented descriptions and known properties of the operational environment within which the safety system will operate. Individual user instructions should not contradict other instructions. Uniform and consistent terminology, notation, and definitions should be used throughout the training manuals.

Style means that the training manual is understandable by the users. A precise definition of each technical term should exist, either in the training manual or in a separate dictionary or glossary. The operations manual may be organized in the style of a tutorial guide, with the assumption that the users also have access to the operations manual.

Traceability means that a forward trace exists between the SRS, the training plan, and the training manual, which shows how each requirement is to be carried out by the users, or carried out automatically by the safety system without user action, and how the results of each requirement are displayed to the users.

B.3.3.9.2 Review Guidance for STMs

The reviewer should determine that the training manuals are both understandable and useful. One method which can be used for this determination is for the reviewer to take the training courses which use these manuals. The training manuals will generally be aimed at either the technician level or the software engineer, and therefore the determination of the understandability and usability of the training manual needs to take into account the intended use. If the reviewer does not have enough experience with training or training documentation to make this determination, the reviewer may require the assistance of other NRC personnel or independent contractor personnel.

B.4 Review Procedures

Reviews are carried out by a combination of inspection and analysis of documents. The adequacy of the computer development process should be reviewed to confirm that software life cycle plans incorporate appropriate commitments, as described in Subsection B.3.1 above. New software, or an unproven development team, will require greater emphasis on the adequacy of the planning phase. A sample of V&V, safety analysis, and configuration management documentation for various life cycle activity groups should be audited to confirm that the developer's life cycle activities have been properly implemented. Subsection B.3.2 above presents specific criteria from which the inspection activities for each specific life cycle activity may be derived. A sample of software design outputs should be reviewed to confirm that they address the functional requirements allocated to the software, and that the expected software development process characteristics are evident in the design outputs. Subsection B.3.3 above describes functional characteristics and software development process characteristics from which the inspection activities for each specific design output may be derived. SRP Appendix 7.0-A contains additional detail on the software review process and the relationship between software reviews and system reviews.

C. REFERENCES

1. ASME NQA-1. “Quality Assurance Requirements for Nuclear Facility Applications.”

2. ASME NQA-1aAddenda, “Addenda to ASME NQA-1-2008, Quality Assurance Requirements for Nuclear Facility Applications.”

3. IEEE Std 279-1971, “Criteria for Protection Systems for Nuclear Power Generating Stations.”
4. IEEE Std 603-1991, “IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations.”

5. IEEE Std 7-4.3.2. “IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations.”

6. IEEE Std 828, “IEEE Standard for Software Configuration Management Plans.”

7. IEEE Std 829, “IEEE Standard for Software Test Documentation.”

8. IEEE Std 830, “IEEE Recommended Practice for Software Requirements Specifications.”

9. ANSI/IEEE Std 1008, “IEEE Standard for Software Unit Testing.”

10. IEEE Std 1012, “IEEE Standard for Software Verification and Validation Plan.”

11. IEEE Std 1028, “IEEE Standard for Software Reviews and Audits.”

12. IEEE Std 1058.1-1991, “IEEE Standard for Software Project Management Plans.”

13. IEEE Std 1058-1998, “IEEE Standard for Software Project Management Plans.”

14. IEEE Std 1061-1998, “IEEE Standard Criteria for Software Quality Metrics Methodology.”

15. IEEE Std 1074, “IEEE Standard for Developing Software Life Cycle Processes.”

16. IEEE Std 1228-1994, “IEEE Standard for Software Safety Plans.”

17. EPRI Topical Report TR-106439, “Guideline on Evaluation and Acceptance of Commercial Grade Digital Equipment for Nuclear Safety Applications,” Electric Power Research Institute, October 1996.

18. NUREG/CR-6101, “Software Reliability and Safety in Nuclear Reactor Protection Systems,” 1993.

19. NUREG/CR-6421, “A Proposed Acceptance Process for Commercial Off-the-Shelf (COTS) Software in Reactor Applications.” March 1996.

20. NUREG/CR-6463, “Review Guidelines on Software Languages for Use in Nuclear Power Plant Safety Systems,” June 1996.

21. RG 1.28, “Quality Assurance Program Criteria (Design and Construction).”

22. RG 1.152, “Criteria for Use of Computers in Safety Systems of Nuclear Power Plants.”

23. RG 1.168, “Verification, Validation, Reviews and Audits for Digital Computer Software used in Safety Systems of Nuclear Power Plants.”

24. RG 1.169, “Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”

25. RG 1.170, “Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”

26. RG 1.171, “Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”

27. RG 1.172, “Software Requirements Specifications for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”

28. RG 1.173, “Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”

29. Safety Evaluation by the Office of Nuclear Reactor Regulation, “EPRI Topical Report TR-106439, “Guideline on Evaluation and Acceptance of Commercial Grade Digital Equipment for Nuclear Safety Applications,” July 17, 1997.

PAPERWORK REDUCTION ACT STATEMENT

The information collections contained in the Standard Review Plan are covered by the requirements of 10 CFR 21, 10 CFR 50 and 10 CFR 73, and were approved by the Office of Management and Budget, approval number 3150-0035, 3150-0011, and 3150-0002.

PUBLIC PROTECTION NOTIFICATION

The NRC may not conduct or sponsor, and a person is not required to respond to, a request for information or an information collection requirement unless the requesting document displays a currently valid OMB control number.

BTP Section 7-14
Description of Changes

BTP 7-14, "Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems"

This BTP Section affirms the technical accuracy and adequacy of the guidance previously provided in BTP 7-14, Revision 5, dated March 2007. See ADAMS Accession Number ML070670183.

The main purpose of this update is to incorporate the revised software RGs and the associated endorsed standards. For organizational purposes, the revision number of each RG and year of each endorsed standard is now listed in one place, Table 7-1. As a result, revisions of RGs and years of endorsed standards were removed from this section, if applicable. For standards that are incorporated by reference into regulation (IEEE Std 279-1971 and IEEE Std 603-1991) and standards that have not been endorsed by the agency, the associated revision number or year is still listed in the discussion.

To be consistent with 10 CFR 73.54 (2009) and RG 1.152 (2011), cyber-security discussions in this section were deleted and discussions of the Secure Development and Operational Environment were added.

Part of 10 CFR was reorganized due to a rulemaking in the fall of 2014. Quality requirement discussions in the former 10 CFR 50.55a(a)(1) were moved to 10 CFR 50.54(jj) and 10 CFR 50.55(i). The incorporation by reference language in the former 10 CFR 50.55a(h)(1) was moved to 10 CFR 50.55a(a)(2). There were no changes either to 10 CFR 50.55a(h)(2) or 10 CFR 50.55a(h)(3).

Additional changes were editorial.

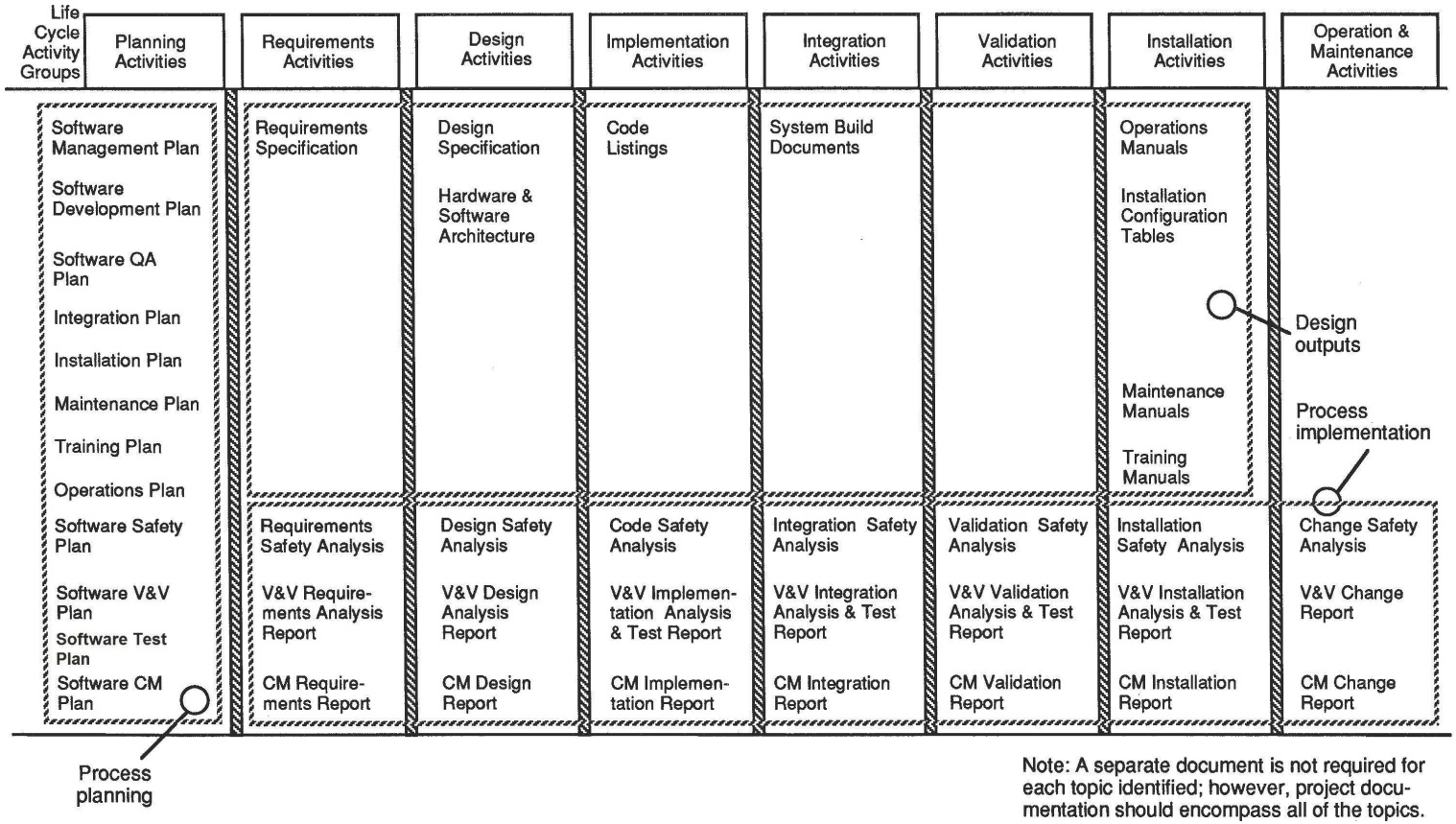

Figure 7-A-1. Flow of Documents Through the Software Life Cycle

BTP 7-14 Numbering Index