WCAP-16096-NP-A
Revision 5.1

May 2021

Software Program Manual for Common Q™ Systems

Westinghouse

Westinghouse Non-Proprietary Class 3

Nuclear Safety Related

Software Program Manual for Common Q™ Systems

WCAP-16096-NP-A

Rev. 5.1

May 2021

APPROVALS

Function	Name and Signature
Author	Matthew A. Shakun* Principal Licensing Engineer, Licensing Engineering
Reviewed	Richard M. Paese* Fellow Engineer, Licensing Engineering
	Jerry M. Stanley* Manager, Safety Software Engineering
	Murat S. Uzman* Fellow Engineer, Independent Verification and Validation
	Roger Costantino* Manager, Independent Verification and Validation
Approved	Stephen L. Packard* Manager, Global I&C Quality & Operations Planning
	Kenneth G. Lunz* Director, Safety & Reactor Systems Engineering
	Anthony J. Schoedel* Manager, Licensing Engineering

*Electronically approved records are authenticated in the electronic document management system.

Table of Contents

Section A

Final Safety Evaluation for "WCAP-16096-P/NP, Revision 5, 'Software Program Manual for Common Q™ Systems'" (ADAMS Accession No. ML18270A029)

Westinghouse acquired the AC160 product line from ABB on April 29th, 2021. Any references in the following Safety Evaluation to ABB's ownership of AC160 is historical in nature.

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555-0001

November 2, 2018

Mr. James A. Gresham, Manager
Regulatory Compliance
Westinghouse Electric Company
1000 Westinghouse Drive
Building 3 Suite 310
Cranberry Township, PA 16066

SUBJECT: FINAL SAFETY EVALUATION FOR "WCAP-16096-P/NP, REVISION 5, 'SOFTWARE PROGRAM MANUAL FOR COMMON QTM SYSTEMS'" (EPID: L-2017-TOP-0059)

Dear Mr. Gresham:

By letter dated August 28, 2017 (Agencywide Documents Access and Management System (ADAMS) Accession No. ML17241A112), Westinghouse Electric Company (Westinghouse) submitted for U.S. Nuclear Regulatory Commission (NRC) staff review Topical Report (TR) "WCAP-16096-P/NP, Revision 5, 'Software Program Manual for Common Q™ Systems.'" By letter dated September 5, 2018, the NRC staff issued its draft safety evaluation (SE) on BWRVIP-41, Revision 4 (ADAMS Accession No. ML18151A486).

By letter dated September 19, 2018 (ADAMS Accession No. ML18269A235), Westinghouse provided comments on the NRC staff draft SE. The comments provided by the Westinghouse were editorial and clarifications.

The NRC staff has found that WCAP-16096, Revision 5 is acceptable for referencing in licensing applications for nuclear power plants to the extent specified and under the limitations delineated in the TR and in the enclosed final SE. The final SE defines the basis for our acceptance of the TR.

Our acceptance applies only to material provided in the subject TR. We do not intend to repeat our review of the acceptable material described in the TR. When the TR appears as a reference in license applications, our review will ensure that the material presented applies to the specific plant involved. License amendment requests that deviate from this TR will be subject to a plant-specific review in accordance with applicable review standards.

In accordance with the guidance provided on the NRC website, we request that Westinghouse publish an accepted version of WCAP-16096, Revision 5 within three months of receipt of this letter. The approved versions shall incorporate this letter and the enclosed final SE after the title page. Also, the accepted version must contain historical review information, including NRC staff requests for additional information (RAIs) and your responses. The approved version shall include an "-A" (designating accepted) following the TR identification symbol.

As an alternative to including the RAIs and RAI responses behind the title page, if changes to the TRs were provided to the NRC staff to support the resolution of RAI responses, and the NRC staff reviewed and accepted those changes as described in the RAI responses, there are two ways that the accepted version can capture the RAIs:

1. The RAIs and RAI responses can be included as an Appendix to the accepted version.
2. The RAIs and RAI responses can be captured in the form of a table (inserted after the final SE) which summarizes the changes as shown in the accepted version of the TR. The table should reference the specific RAIs and RAI responses which resulted in any changes, as shown in the accepted version of the TR.

If future changes to the NRC's regulatory requirements affect the acceptability of this TR, Westinghouse will be expected to revise the TR appropriately. Licensees referencing this TR would be expected to justify its continued applicability or evaluate their plant using the revised TR.

If you have any questions or require any additional information, please feel free to contact the NRC Project Manager for the review, Joseph Holonich at (301) 415-7297 or joseph.holonich@nrc.gov.

Sincerely,

/RA/

Dennis C. Morey, Chief
Licensing Processes Branch
Division of Policy and Rulemaking
Office of Nuclear Reactor Regulation

Docket No. 99902038

Enclosure:
Final Safety Evaluation

SUBJECT: FINAL SAFETY EVALUATION FOR "WCAP-16096-P/NP, REVISION 5, 'SOFTWARE PROGRAM MANUAL FOR COMMON QTM SYSTEMS,'" (EPID: L-2017-TOP-0059) DATED NOVEMBER 2, 2018

DISTRIBUTION:

PUBLIC	RidsACRS_MailCTR	RidsNrrLADHarrison	RidsNrrDlpPrlb
RidsNrrDeEicb	RidsNrrDlp	RidsResOd	RidsNroOd
RidsNrrDlr	RidsOgcMailCenter	MWaters, NRR	RStattel, NRR
DTaneja, NRO	WRoggenbrodt, NRO	RidsNrrDe	JHolonich, NRR
DMorey, NRR	PLPB r/f		

ADAMS Accession No.: ML18270A029 *via e-mail NRR-106

OFFICE	NRR/DLP/PLPB/PM*	NRR/DLP/PLPB/LA*	NRR/DE/EICB/BC*
NAME	JHolonich	DHarrison	MWaters (RAlvarado for)
DATE	11/02/2018	10/31/2018	10/01/2018

OFFICE	NRO/DEI/ICE/ABC*	NRR/DLP/PLPB/BC
NAME	DTaneja	DMorey
DATE	10/01/2018	11/02/2018

OFFICIAL RECORD COPY

U.S. NUCLEAR REGULATORY COMMISSION STAFF

SAFETY EVALUATION FOR

WESTINGHOUSE TOPICAL REPORT WCAP-16096-P, REVISION 5,

"SOFTWARE PROGRAM MANUAL FOR COMMON Q SYSTEMS"

November 2018

Principal Contributors:

Rich Stattel

William Roggenbrodt

Enclosure

TABLE OF CONTENTS FOR COMMON Q SAFETY EVALUATION

U.S. NUCLEAR REGULATORY COMMISSION STAFF

SAFETY EVALUATION FOR

WESTINGHOUSE TOPICAL REPORT WCAP-16096-P, REVISION 5,

"SOFTWARE PROGRAM MANUAL FOR COMMON Q SYSTEMS"

1.0 INTRODUCTION

The Software Program Manual (SPM) for Common Qualified (Common Q) Systems was originally submitted as document CE-CES-195-P by Combustion Engineering (CE), for U.S. Nuclear Regulatory Commission (NRC) staff review in 2000. Subsequently, the commercial nuclear power businesses of Asea Brown Boveri (ABB), of which CE was a part, were purchased by British Nuclear Fuels Limited (BNFL) and eventually integrated into the Westinghouse Electric Company (WEC), such that the SPM is now owned by WEC. See References 10 and 11 for Revision 1 of this document and the associated safety evaluation (SE). This document specifies the life cycle planning process for Common Q application software. The SPM specifies the development, documentation, utilization, and maintenance of software to be developed for use with the Common Q platform in nuclear safety applications. It also provides guidance for the maintenance, implementation, and use of commercial-grade hardware and previously developed software (PDS). Revision 4 of the Common Q SPM was submitted by WEC (Refs. 3 and 4) and approved by the NRC (Ref. 17).

The SPM is being updated to Revision Level 5 per Reference 14 to include a revised test approach that defines testing requirements for Nth of a kind systems of the same design. The revised SPM also addresses corrective actions, implements process improvements, updates several of its references, and includes other minor changes.

The SPM specifies procedures and controls for the complete software development process. This process includes the integration of software into system hardware. Since the application software has not yet been developed, the staff's evaluation does not include the review of the implementation or outputs of the life cycle process, but is limited to the evaluation of the specified planning processes.

2.0 REGULATORY EVALUATION

2.1 Regulatory Criteria

The following regulatory requirements are applicable to the review of the Common Q SPM.

Title 10 of the *Code of Federal Regulations* (10 CFR)

- 10 CFR 50.55a(a)(1) requires, in part, that systems and components be designed, tested, and inspected to quality standards commensurate with the safety function to be performed.

- 10 CFR 50.55a(h), "Protection and Safety Systems," requires compliance with 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 dated January 30, 1995.

  - Clause 5.3 of IEEE Std. 603-1991 requires that components and modules shall be of a quality that is consistent with minimum maintenance requirements and low failure rates. It also requires that safety system equipment be designed, manufactured, inspected, installed, tested, operated, and maintained in accordance with a prescribed quality assurance program.

  - Clause 5.6.3 of IEEE Std.603-1991 requires safety system to be designed such that credible failures in and consequential actions by other systems will not prevent safety systems from performing their intended safety functions.

  - Clause 5.9 of IEEE Std. 603-1991 requires the design to permit the administrative control of access to safety system equipment. These administrative controls shall be supported by provisions within the safety systems, by provision in the generating station design, or by a combination thereof.

- 10 CFR Part 50, Appendix A, General Design Criteria (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.

- 10 CFR Part 50, Appendix A, GDC 21 requires, in part, that protection systems must be designed for high functional reliability commensurate with the safety functions to be performed.

- 10 CFR Part 50, Appendix B, “Quality Assurance Criteria for Nuclear Power Plants and Fuel Processing Plants”, Criterion I, “Organization,” requires in part that the applicant shall be responsible for the establishment and execution of the quality assurance program.

- 10 CFR Part 50, Appendix B, Criterion II, “Quality Assurance Program,” requires in part that the applicant shall establish at the earliest practicable time, consistent with the schedule for accomplishing the activities, a quality assurance program which complies with the requirements of Appendix B.

- 10 CFR Part 50, Appendix B, Criterion III, “Design Control,” requires, in part that, for safety-related structures systems, or components (SSCs), quality standards be specified and that design control measures shall provide for verifying or checking the adequacy of design.

- 10 CFR Part 50, Appendix B, Criterion V, “Instructions, Procedures, and Drawings,” requires, in part that, for safety-related SSCs, activities affecting quality shall be prescribed by documented…procedures…of a type appropriate to the circumstances….

- 10 CFR Part 50, Appendix B, Criterion VI, “Document Control,” requires, in part that, for safety-related SSCs, measures shall be established to control the issuance of documents which prescribe all activities affecting quality.

- 10 CFR Part 50, Appendix B, Criterion VII, “Control of Purchased Material and Services,” requires documented control of purchased material, equipment, and services for safety-related SSCs.

- 10 CFR Part 50, Appendix B, Criterion XI, “Test Control,” requires, in part, that a test program be established to demonstrate that safety-related systems and components will perform satisfactorily in service.

- 10 CFR Part 50, Appendix B, Criterion XV, “Nonconforming Materials, Parts, or Components” requires in part that measures shall be established to control materials, parts, or components which do not conform to requirements in order to prevent their inadvertent use or installation.

The following guidance documents are applicable to, and were utilized in support of, the review of the Common Q Software Program Manual.

Regulatory Guides (RGs)

- RG 1.152, Revision 3, “Criteria for Use of Computers in Safety Systems of Nuclear Power Plants.”

- RG 1.168, Revision 2, “Verification, Validation, Reviews and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”

- RG 1.169, Revision 1, “Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”

- RG 1.170, Revision 1, “Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”

- RG 1.171, Revision 1, “Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”

- RG 1.172, Revision 1, “Software Requirements Specifications for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”

- RG 1.173, Revision 1, “Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants.”

NUREG-Series Publications

NUREG-0800, Revision 7, “Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants: LWR Edition,” Chapter 7, “Instrumentation and Controls,” March 2007.

- Branch Technical Position (BTP) 7-14, Revision 6, “Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems.”

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

Industry Standards

- IEEE Std. 7-4.3.2-2003, “Application Criteria for Programmable Digital Computer Systems in Safety Systems of Nuclear Power Generating Stations,” as endorsed by RG 1.152.

- IEEE Std. 730-1998, “Software Quality Assurance Plans,”

- IEEE Std. 828-2005, “Software Configuration Management Plans,” as endorsed by RG 1.169.

- IEEE Std. 829-1983, “Software Test Documentation,” as endorsed by RG 1.170, September 1997.

- IEEE Std. 829-1998, “Software Test Documentation,”

- IEEE Std. 830-1998, “Guide for Software Requirements Specifications,” as endorsed by RG 1.172.

- IEEE Std. 1008-1987 (Reaffirmed 2009), “IEEE Standard for Software Unit Testing.”

- IEEE Std. 1012-2004, “IEEE Standard for Software Verification and Validation Plans,” as endorsed by RG 1.168.

- IEEE Std. 1028-2008, “IEEE Standard for Software Reviews and Audits,” as endorsed by RG 1.168.

- IEEE Std. 1042-1987, “IEEE Guide to Software Management.”

- IEEE Std. 1063-2001, “IEEE Standard for Software Documentation.”

- IEEE Std. 1074-2006, “IEEE Std. for Developing Software Life Cycle Processes,” as endorsed by RG 1.173.

2.2 Method of Review

The staff used the guidance in RGs and BTP 7-14 to review the software life cycle plans outlined in the Common Q SPM. In BTP 7-14 the information to be reviewed is subdivided into the following three topic areas:

- Software life cycle process planning;
- Software life cycle process implementation; and
- Software life cycle process design outputs.

2.3 Precedents

The NRC previously evaluated the Common Q SPM which was submitted by WEC as document number WCAP-16096-P/NP-A, Revision 4 and the results of this evaluation are documented in the associated SE (Ref. 4).

3.0 TECHNICAL EVALUATION

The regulation at 10 CFR Part 50.55a(a)(1) requires, in part, that systems and components be designed, tested, and inspected to quality standards commensurate with the safety function to be performed. The regulation at 10 CFR Part 50, Appendix A, GDC 1 requires, in part, that a quality assurance program be established and implemented in order to provide adequate assurance that systems and components important to safety will satisfactorily perform their safety functions. The regulation in 10 CFR Part 50, Appendix B, describes criteria that a quality assurance program for systems and components that prevent or mitigate the consequences of postulated accidents must meet. In particular, besides the systems and components that directly prevent or mitigate the consequences of postulated accidents, the criteria of Appendix B also apply to all activities affecting the safety-related functions of such systems and components as designing, purchasing, installing, testing, operating, maintaining, or modifying.

BTP 7-14, provides an acceptable way to meet the regulations cited. The staff reviewed the Common Q SPM in accordance with BTP 7-14.

Acceptability of software for safety system functions is dependent upon (1) confirmation that acceptable plans were prepared to control software development activities as described in BTP 7-14, B.3.1, (2) evidence that the plans were followed in an acceptable software life cycle as described in BTP 7-14, B.3.2, and (3) evidence that the process produced acceptable design outputs as described in BTP 7-14, B.3.3. The Common Q SPM only addresses the first item, the planning phase.

This SE instructs applicants referencing Topical Report WCAP-16096-P (NP), Revision 5 (Ref. 14) to make available specified information. The meaning of the term "make available," however, depends on the type of application referencing the topical report, as follows: A licensee requesting amendment of an existing operating license will make available the identified information by including it in the application. An applicant for certification of a standard design will make available the identified information at the time of presentation of the application or by proposing Inspections, Tests, Analyses, and Acceptance Criteria (ITAAC) that address it. Similarly, an applicant for a Combined License (COL) will make available the identified information by providing the necessary information at the time of license application or by (1) proposing ITAAC or by referencing a certified design that does so and (2) addressing any remaining COL action items identified in connection with the topical report in the design certification. A COL holder will ultimately address the information through the process of closing the associated ITAAC if any have been utilized during the licensing process.

BTP 7-14, A.3.1, describes three software planning characteristics: management, implementation, and resource. Management characteristics are significant to the management of the project activities. Implementation characteristics describe the work necessary to achieve the purpose of the planning documents. Resource characteristics describe the material resources necessary to carry out the work defined in the planning document. The Common Q SPM was reviewed against these planning characteristics. These characteristics were assessed and compared to the characteristics described in BTP 7-14 to determine the adequacy of software planning activities implemented for Common Q.

3.1 Design Considerations

The Common Q platform is a distributed, microprocessor-based computer system. It is capable of being configured with three or four independent redundant data-processing paths or divisions,

each with two or three layers of operation. Data processing paths can be run asynchronously with respect to each other. Layers of operation include signal acquisition, data-processing, and actuation signal voting. The Common Q platform uses microprocessor-based digital equipment, operating system software, and plant-specific application software to perform safety-related I&C system functions at nuclear power plants. A full description of the Common Q platform may be found in the Common Q platform TRs (Refs. 1 and 14).

Application software is developed for project-specific applications of the Common Q platform. Software implements plant-specific I&C control and logic functions, and is hardware dependent. Software will be developed using WEC approved software development tools. The Common Q SPM describes the conditions and objectives to develop application software.

3.2 Life Cycle Planning Process for Application Software

Digital Instrumentation and Control (I&C) safety systems must be designed, fabricated, installed, and tested to quality standards commensurate with the level of the importance of the safety functions to be performed. The development of safety system software should progress according to a formally defined software lifecycle (SLC). Implementation of an acceptable SLC provides reasonable assurance the necessary software quality has been instilled in the final system. BTP 7-14, Section B.2.1 states that the information to be reviewed for the software life cycle process planning should be found under the following topics:

B.3.1.1 Software Management Plan
B.3.1.2 Software Development Plan
B.3.1.3 Software Quality Assurance Plan
B.3.1.4 Software Integration Plan
B.3.1.5 Software Installation Plan
B.3.1.6 Software Maintenance Plan
B.3.1.7 Software Training Plan
B.3.1.8 Software Operations Plan
B.3.1.9 Software Safety Plan
B.3.1.10 Software Verification and Validation Plan
B.3.1.11 Software Configuration Management Plan
B.3.1.12 Software Test Plan

In addition, WEC developed a separate Secure Development and Operating Environment (SDOE) plan to address the criteria of RG 1.152 which provides guidance for the establishment of a SDOE for safety related software. Section 12 of the SPM constitutes the Common Q SDOE Plan.

While most of the information about the above topics is in the SPM, information found in the other submittals and in previous revisions of the SPM is sometimes helpful to the evaluation, and therefore, was considered for this evaluation. The SPM includes sections with the following section numbers and titles:

- (Section 3) Software Safety Plan (SSP)
- (Section 4) Software Quality Assurance Plan (SQAP)
- (Section 5) Software Verification and Validation Plan (SVVP)
- (Section 6) Software Configuration Management Plan (SCMP)
- (Section 7) Software Test Plan (STP)

- (Section 8) Software Installation Plan (SIP)
- (Section 9) Software Maintenance Plan (SMP)
- (Section 12) Secure Development and Operational Environment Plan

The staff found the information needed to support its safety conclusions on the balance of the life cycle topics either in the balance of the SPM or in the Common Q TR WCAP-16097-P "Common Qualified Platform" (Ref. 14) and its appendices. The staff has organized this report to follow the sequence outlined under the topic in BTP 7-14. BTP 7-14, Section B.3.1 describes the acceptance criteria used for reviewing the 12 software plans of the SPM.

3.2.1 Software Management Plan

The Software Management Plan (SMP) describes the management aspects of the software development project. BTP 7-14, Section B.3.1.1 describes acceptance criteria for software management plans. RG 1.173 endorses IEEE Std. 1074-2006, "IEEE Standard for Developing Software Life Cycle Processes." IEEE Std. 1074-2006 describes, in terms of inputs and outputs, a set of processes and constituent activities that are commonly accepted as comprising a controlled and well-coordinated software development process. IEEE Std. 1074-2006 Annex A, Section A.1, "Project Management Section of Activity Groups," describes an acceptable approach for software project management. It states that project management processes are, "the processes that initiate, monitor, and control software projects throughout the software life cycle."

The required elements of a Software Management Plan are contained within Sections 2, 4.3, 5.5.1, and 6.2 of the Common Q SPM. These sections of the SPM define a strategy for managing Common Q software projects. Each of these sections was reviewed against the specific acceptance criteria established by BTP 7-14.

Section 4.3 of the Common Q SPM describes the management principles used for the development of Common Q application software for each phase of the software development life cycle. It includes a description of the software project planning organization which includes a general overview of the organizational structure used by WEC and a discussion of the responsibilities that each of the following organizations has within the Nuclear Automation Organization.

- Quality Organization
- Engineering Organization
  - Design Team
  - V&V Team

The specific tasks and responsibilities performed by these organizations during each of the software lifecycle phases are described within the SPM. These tasks include software design and development, software quality assurance planning, verification reviews, audits, test planning, test execution, and test reporting. The SPM describes the interfaces and boundaries that exist between these organizations.

A level of independence between the Verification and Validation (V&V) Team and the Design Team is established by specifying different reporting structures up to the director level. Beyond the director level, the two teams report to the same vice president. The directors to which the V&V team and the Design team report are administratively and financially independent of one another. This relationship between the design team and the independent verification and

validation (IVV) team is illustrated in Exhibit 2-1, “Design/IV&V [independent verification and validation] Team Organization,” of the SPM. The degree of independence between the V&V team and the design team is further reinforced by not allowing V&V team members to participate on the design team.

The SPM calls for the development of a project specific Project Quality Plan (PQP) during the Initiation (Concepts) Phase of the software development life cycle. The PQP allows for alternatives to the SPM processes. Because of this, the PQP should be reviewed to determine if the justification for the use of alternatives to the SPM or other, additional metrics or qualifiers beyond the directions within the SPM is acceptable when an applicant requests approval for installation of a safety-related system based on the Common Q platform. This is plant specific action item 1.

Per BTP 7-14, Sections B.3.2 and B.3.3, the implementation activities and design outputs are to be separately evaluated so that the application design can be evaluated to determine that the software management plan has been followed. This is plant specific action item 2.

The elements of the software management plan are incorporated into the Common Q SPM. The staff has reviewed the Common Q SPM and finds that it establishes adequate organization and authority structure for the design, the procedures to be used, and the relationships between major activities. The staff finds that the management structure in the Common Q SPM provides for adequate project oversight, control, reporting, review, and assessment. The management structure also supports independence of V&V activities. The staff concludes that the Common Q SPM meets the requirements for a software management plan as outlined in IEEE Std. 1074-2006 as endorsed by RG 1.173 and, is, therefore, acceptable.

3.2.2 Software Development Plan

The Software Development Plan (SDP) describes the plan for technical project development. BTP 7-14, Section B.3.1.2 describes acceptance criteria for software development plans. RG 1.173 endorses IEEE Std. 1074-2006 as providing an acceptable approach to software development processes. BTP 7-14 states that the SDP should clearly state tasks of each life cycle, and state the life cycle inputs and outputs. The review, verification and validation of those outputs should be defined. IEEE Std. 7-4.3.2-2003 provides additional guidance on software development processes.

WEC uses a controlled software development process which is defined within the Common Q SPM. The criteria for the Common Q software development plan are satisfied by a project plan and a Project Quality Plan. These plans are created for each Common Q project in accordance with general criteria that is defined within the SMP. The required elements of a Software Development Plan are defined within the following SPM sections:

- 1.2.1, “Software Classification and Categorization”
- 1.4.1, “Software Life Cycle”
- 4.1.3, “Software Development Process”
- 5.9, “Software Integrity Level Scheme”

Common Q Software Life Cycle

Section 1.4.1 of the SPM defines the software lifecycle (SLC) used for the development of Common Q software. This life cycle is consistent with a classic waterfall model like the model discussed in Section 2.3.1 of NUREG/CR-6101. The Common Q SLC consists of the following life cycle phases:

- Concept
- Requirements Analysis
- Design
- Implementation or Coding
- Test
- Installation and Checkout
- Operation and Maintenance
- Retirement

This model assumes that each phase of the life cycle is completed in sequential order from concept to the retirement phase. The staff finds the WEC choice of SLC acceptable since the waterfall model is well suited for projects with known and stable requirements and where few changes to requirements are anticipated. Since WEC selected an acceptable software life cycle model, the guidance criteria of IEEE Std. 1074-2006, Section A.1 has been satisfied.

Common Q Software Life Cycle Tasks (Inputs & Outputs)

BTP 7-14, Section B.3.1.2.4 states that an applicant should identify which tasks are included with each life cycle phase, and identify the life cycle tasks' inputs and outputs. Exhibit 4-3 of the SPM identifies tasks which are performed for various software categories (defined by the Common Q software integrity scheme described below) during the SLC process and identifies the phases during which each task is performed. Revision 5 of the SPM adds tasks to accommodate the System Validation Testing and Factory Acceptance Testing in accordance with the updated test methods presented in the SPM. In addition, Exhibit 5-1, "Software Tasks and Responsibilities," of the SPM defines the responsibilities for completion of software tasks.

Note: Several exhibits are included in the SPM to show that all required V&V tasks are included as part of the SLC processes. In Exhibits 4-3 and 5-1, WEC has grouped individual tasks into general category headings. For example the task "Design Verification" may include several individual subtasks that are not listed in Exhibit 5-1. As such, specific individual V&V tasks are not delineated in these tables. Exhibit 5-8 was created in conjunction with Section 5 of the SPM to list and define the specific V&V tasks and to map these tasks to the V&V activities defined within IEEE Std. 1012-2004. Exhibit 5-8 was updated in SPM Revision 5 to accommodate System Validation Testing and Factory Acceptance Testing in accordance with the updated test methods presented in the SPM.

IEEE Std. 1012-2004, Clause 1.7, "Conformance," states that the minimum V&V tasks are defined by the software integrity level assigned to the software. Exhibit 5-8 of the SPM includes a table which identifies the minimum tasks for each software integrity level of the Common Q platform. This exhibit contains a mapping of the V&V activities associated with the development lifecycle of a Common Q system to the IEEE Std. 1012-2004 standard. This mapping table also identifies the phase of the development lifecycle in which each activity is performed. Several V&V activities are performed multiple times during the development process. The left-hand

column of this table lists all of the V&V activities from Table 2 of IEEE Std. 1012-2004. Each of these activities has a corresponding activity and reference to the SPM section for the equivalent activity within the Common Q development process. The staff reviewed the activities included in this mapping table and determined that it contains sufficient detail and reference to the SPM to show that the V&V activities performed for safety related Common Q application Protection software are consistent with high criticality software developed to software integrity level (SIL) Level 4 as defined by IEEE Std. 1012-2004 and is therefore acceptable.

Common Q Software Integrity Level Scheme

Section 5.9 of the Common Q SPM discusses the WEC Common Q specific software classification or software integrity level scheme.

Table 5.9.1 of the SPM compares the WEC software integrity level scheme with the scheme presented within IEEE Std. 1012-2004. IEEE Std. 1012-2004 states: "This standard uses software integrity levels to determine the V&V tasks to be performed. High-integrity software requires a larger set of V&V processes and a more rigorous application of V&V tasks." Section 1.2.1 of the SPM defines the software classes used for Common Q software as follows:

- ***Protection*** *(safety critical). Software whose function is necessary to directly perform RPS control actions, ESFAS control actions, and safe shutdown control actions.*

- ***Important-to-Safety****. Software whose function is necessary to directly perform alternate protection system control actions or software that is relied on to monitor or test protection functions, or software that monitors plant critical safety functions.*

- ***Important-to-Availability****. Software that is relied on to maintain operation of plant systems and equipment that are critical to maintaining an operating plant.*

- ***General Purpose****. Software that performs some purpose other than that described in the previous classifications. This software includes tools that are used to develop software in the other classifications, but is not installed in the online plant system. Examples of General Purpose software include commercial grade dedication test software, compilers, assemblers, linkers, comparators, editors, test case generators, and test coverage analyzers.*

Exhibit 4-1 of the SPM identifies assignment of Common Q components to the software classes described above. All Common Q application software on the Advant Controller 160 (AC160) safety processors, the Operator Modules (OM's) and the Maintenance and Test Panels (MTP's) are classified as either Protection, which is equivalent to SIL 4 as defined in IEEE 1012-2004, or Important to Safety. This is consistent with the fact that Common Q system is classified as Class 1E as defined by IEEE Std. 603-1991.

Common Q Components and software that are classified as either Protection or Important to Safety are considered to be safety related. It is however, understood that the subset of safety related software that is classified as Important to Safety does not directly perform RPS or ESFAS safety functions. For this reason, it is acceptable for Important to Safety software to be developed using V&V activities that are not equivalent to SIL Level 4 activities as defined in IEEE Std. 1012-2004.

The staff finds the software integrity level scheme used for the Common Q platform and application development acceptable since it is similar to the software integrity level scheme defined in IEEE Std. 1012-2004, and because the scheme is appropriately used to establish a minimum set of V&V tasks for development of Common Q application software. Section 3.2.10 of this SE provides additional evaluation of the V&V tasks performed on Common Q software.

Management and Oversight of the Software Development Processes

The project manager is responsible for ensuring that the design, verification and validation, and quality assurance (QA) activities are conducted in accordance with the SPM. The corrective action program used during the Common Q development process is defined in Section 11, "Problem Reporting and Corrective Action," of the SPM. This program is designed to promptly identify and correct conditions adverse to safety and quality. This program provides oversight to ensure that development process will be followed and any deviation will be discovered in time to take corrective action. This section of the SPM was updated to accommodate the changed testing processes being implemented within the SPM and to clarify use and management aspects of the corrective action program associated with Revision 5 of the SPM. Also, Exhibit 11-2 was eliminated from the SPM. This exhibit had been a sample printout of a software tool used to implement the corrective action processes. Required information for exception reporting is now captured in Exhibit 11-1 and specific tool usage information is being omitted. The NRC staff considers this acceptable as long as the minimum required information for exception reporting is retained.

Software Tools

BTP 7-14, Section B.3.1.2.4 provides guidance for software tools, and references IEEE Std. 7-4.3.2-2003, Clause 5.3.2, which states, in part, that software tools used to support software development processes and verification and validation processes shall be controlled under configuration management. To confirm the software tools are suitable for use, the clause further states either a test tool validation program shall be developed to provide confidence that the necessary features of the software tool function as intended or the software tool shall be used in a manner such that defects not detected by the software tool will be detected by V&V activities.

The Common Q SPM Sections 3.3.10, "Tool Support and Approval," and 4.9, "Tools, Techniques and Methodologies," discuss the development support tools used to facilitate Common Q application software development. An evaluation of a tool's readiness for use on a project is performed before such a tool is used to support the development of a Common Q application. This evaluation considers; the tool's past performance, extent of tool validation performed, consistency of tool design with planned use, use of tool upgrades, retirement of the tool, and restrictions on the use of the tool due to its limitations. The configuration management, software quality assurance and IVV processes defined within the SPM apply to software tools and provide a means of ensuring that these tools are only used for their approved and intended purposes. The outputs of software tools undergo the V&V process as defined in the Software Verification and Validation Plan (SVVP), in SPM Section 5.

The staff has reviewed the Common Q SPM and concludes that the software development plan conforms with the criteria provided by IEEE Std. 1074-2006, "IEEE Standard for Developing Software Life Cycle Processes," as endorsed by RG 1.173, "Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants." In addition, the SPM adequately addresses the software development planning activities of

BTP 7-14. The SPM describes acceptable methods of organizing the software life cycle. The staff, therefore, concludes that WEC's application software development plan is acceptable.

3.2.3 Software Quality Assurance Plan

BTP 7-14, Section B.3.1.3 provides guidance in evaluating a Software Quality Assurance Plan (SQAP). The SQAP shall conform to the requirements of 10 CFR Part 50, Appendix B, and the applicant's overall QA program. 10 CFR Part 50, Appendix B states that the applicant shall be responsible for the establishment and execution of the quality assurance program. The applicant may delegate the work of establishing and executing the quality assurance program, or any part thereof, but shall retain responsibility for the quality assurance program. The SQAP would typically identify which QA procedures are applicable to specific software processes, identify particular methods chosen to implement QA procedural requirements, and augment and supplement the QA program as needed for software.

IEEE Std. 7-4.3.2-2003, Clause 5.3.1, which is endorsed by RG 1.152 provides guidance on software quality assurance. IEEE Std.7-4.3.2-2003, Clause 5.3.1, states, "Computer software shall be developed, modified, or accepted in accordance with an approved software QA plan."

The Common Q SQAP for application software is described in Section 4 of the SPM, "Software Quality Assurance Plan." The SQAP describes the methodology used for managing Common Q software throughout the development life cycle. Section 4.1.1 of the SPM states that the Common Q SPM complies with IEEE Std. 730-1998. The scope of the Common Q SQAP includes software in all four SIL classifications: protection, important to safety, important to availability, and general purpose. The Common Q SQAP applies to original protection and important to safety software that was developed under the requirements of the Common Q SPM.

Evaluations of existing software not created under the controls of the Common Q SPM are performed in order to qualify this software for use under the Common Q SPM. For commercial software, qualification is achieved through the use of WEC's commercial grade dedication program. For non-commercial protection and important to safety software that has actively been used in a nuclear power plant being implemented in Common Q, an evaluation is performed to ensure the quality assurance program being used for development and maintenance of this software is acceptable and includes the following:

- The effective quality assurance program has an active program for problem and corrective action reporting.
- The software has adequate design documentation.
- The software has adequate user documentation.
- The software includes well commented source code.
- The software has been verified and validated under a program that the IVV team determines to be appropriate.

For non-commercial software that has not been actively used in a nuclear power plant being implemented in Common Q, an evaluation is performed to ensure that appropriate quality controls commensurate with the safety classification of the software are implemented.

Quality assurance tasks are listed in Exhibit 4-3 of the SPM. These quality assurance tasks are described in Section 4 of the SPM for each software life cycle phase. These descriptions

include a discussion of the tasks and the responsibilities of the organizations performing software quality assurance activities. In addition, Exhibit 5-1 identifies organizational responsibilities for performance of specific software SQA tasks.

Documentation requirements for performance of software Quality Assurance (SQA) activities are described in Section 4.4, "Documentation," of the SMP. Many of the tasks listed in Exhibit 4-3 are in fact documents that will provide evidence for completion of the associated SQA tasks. Furthermore, Section 10 of the SPM, "Documentation," provides guidance for how these documents will be developed.

SPM Section 4.5 identifies the standards, practices, conventions and metrics used for the development of a Common Q based system. It states that, "*compliance with the WEC quality management system standards shall be monitored and assured through the review and audit process."* Standards used for development of Common Q systems include Coding Standards, Software Testing Standards, and Documentation Standards. Coding standards are not established at a generic level and are instead defined within the project specific PQP. Testing standards are defined by the Software Test Plan which is evaluated in Section 3.2.12 of this SE. Documentation Standards are identified in Section 10 of the SPM and include IEEE Std. 830-1998 for Software Requirement Specification (SRS) documentation requirements, IEEE Std. 1016-1998 (Reaffirmed in 2009) for Software Design Description (SDD) documentation requirements, IEEE Std. 1012-2004 for V&V documentation requirements, and IEEE Std. 1063-2001 for Software User documentation requirements.

SPM Section 4.6 describes how software reviews are performed for Common Q applications. Software reviews are performed to verify technical adequacy and to verify completeness of the design and development of Common Q software. The SPM lists several software review activities and defines groups responsible for performance of these activities. The following types of reviews which are defined in IEEE Std. 1028-2008 are performed for Common Q software developed under the SPM:

- Management Reviews,
- Technical Reviews,
- Inspections,
- Walk-through's, and
- Audits.

SPM Section 4.6.2 describes the minimum software reviews and audits to be performed for Common Q software. The staff has determined that this minimum set of review and audit requirements complies with the criteria of IEEE Std. 730-1998 Sections 4.6.2.1 through 4.6.2.10.

IEEE Std.730-1998, Section 4.8 states that the SQAP should describe practices and procedures to be followed for reporting, tracking, and resolving problems. It also stipulates that the SQAP should state specific organizational responsibilities concerned with implementation.

The Common Q SPM Section 11, "Problem Reporting and Corrective Actions," discusses the Common Q processes relating to these criteria. The SPM describes the problem reporting process used to handle discrepancies, deficiencies, or comments identified as a result of testing, review, or other means. The SPM describes two processes used for reporting errors. One is used for errors identified during the development process prior to approval for use in a

nuclear power plant application. The other is used for reporting of errors that are identified after the software has been approved for use. These processes include noncompliance reporting in accordance with 10 CFR Part 21, “Reporting of Defects and Noncompliance.” Organizational responsibilities associated with the problem reporting and corrective action processes are also defined in the SPM.

During the Initiation (Concept) phase, the SQAP calls for the development of a PQP which becomes the operative plan for a specific application development process. This PQP may deviate from the SQAP processes defined in Section 4 of the SPM; however, any such deviations must be documented and justified within the PQP. Because such deviations cannot be evaluated during this safety evaluation, a plant specific action item for evaluating these changes has been created. This is plant specific action item 1.

The regulation at 10 CFR Part 50, Appendix B, allows applicants or licensees to delegate the work of establishing and executing the Quality Assurance program, but applicants/licensees shall retain overall responsibility and shall determine if the quality of the software is sufficient. Applicants or licensees referencing this topical report are to make available a SQAP to address these licensee specific responsibilities. This is plant specific action item 3.

The SQAP stipulates that the SQA organization shall participate in formal reviews and audits of the software development activity. Required reviews and audits are indicated in the plan including review documentation requirements, evaluation criteria, anomaly reporting, and anomaly resolution procedures. Additional reporting of the staff’s evaluation of the SQAP is detailed in Section 3.2.10, "Software Verification and Validation Plan."

The SQAP describes the process by which WEC manages software and documentation throughout the Common Q software development life cycle, and the SQAP conforms to IEEE Std. 730-1998. The Engineering Project Manager is responsible for ensuring all design team activities are performed in accordance with the QA processes and procedures. The SQAP adequately addresses the software quality planning activities of BTP 7-14. The staff concludes that the Common Q SQAP meets the Guidance in BTP 7-14 Section B.3.1.3 with regard to QA software reviews and audits and is, therefore, acceptable.

Revision 5 to the SPM includes a change to the process for development of a site test plan. This change allows development of the site test plan to occur at a later stage of the development lifecycle to support evaluation of requirement testability on-site. The V&V activity for system V&V test plan generation described in Exhibit 5-8 of the SPM was also revised to facilitate later stage development of the site test plan if necessary. The NRC staff finds this change acceptable because required V&V activities are retained. This change allows for later stage completion of required tasks and does not alter the requirements for task completion.

3.2.4 Software Integration Plan

BTP 7-14, Section B.3.1.4 provides guidance in evaluating a Software Integration Plan (SIntP). IEEE Std. 1074-2006, Clause A.1.2.8, “Plan Integration,” which is endorsed by RG 1.173, provides an acceptable approach to an integration plan. Clause A.1.2.8.2 states that during the plan integration activity, the software requirements and the software design description are analyzed to determine the order of combining software components into an overall system. In addition, Clause A.1.2.8.2 states that the integration planned information shall be coordinated with the evaluation planned information. BTP 7-14, Section B.3.1.4.1 guidance calls for a

general description of the software integration process and of the software integration organization.

For the Common Q, WEC does not define a separate software integration organization to perform system integration related activities. Instead, such activities are allocated to different organizations involved with the Common Q software development processes. This allocation of integration activities is defined within various sections within the SMP. For example, Integration Tests are defined in Section 7.3.1.3 of the SPM and Exhibit 5-1 shows that the IVV Team has the responsibility for performing Integration tests for Protection software. Conversely, the design team has the responsibility for performing Integration tests for Important to Availability software.

The testing aspects of Common Q Software Integration are described in Section 7, "Software Test Plan," of the SPM. The Common Q software testing process includes Integration Tests that are conducted on the production hardware or with a system that is functionally equivalent to the production system. This section also specifies that a functionally equivalent system entails a test bed which provides a functionally equivalent configuration to the production hardware.

The NRC staff notes this is a deviation from the integration test description provided in the previous version of the SPM which stated that integration tests were to be performed on actual production hardware. The NRC staff determined that allowing performance of integration tests on non-production hardware is acceptable based on the fact that first of a kind systems undergo system validation tests, which per Section 4.7 of the SPM, encompass the scope of a factory acceptance test (FAT) and subsequent factory tests must still be performed using actual production equipment. Section 7.3.1.5, "Factory Acceptance Test (FAT)," of the SPM states that "FAT includes tests that are performed on the deliverable system for each deliverable system." In addition, Westinghouse confirmed in its response to RAI 7, and RAI 8 (Reference 16) that "… the Factory Acceptance Test (FAT) is performed on the delivered equipment." Subsection 7.3.1.3, "Integration Test," describes the details of the integration tests performed during the development of a Common Q application.

Revision 5 of the SPM changed Section 7.3.1.3, "Integration Test," of the SPM such that the following Integration Test Items listed were removed.

- Error Handling
- Communications
- Redundancy
- Diversity

In response to RAI 6 (Ref. 15), Westinghouse stated that because integration testing is used as part of system validation testing when validating the design and as part of the FAT testing to demonstrate the deliverable system has been properly integrated, the removed test items will continue to be performed and are included as test items in Sections 7.3.1.4, "System Validation Test," and 7.3.1.5, "Factory Acceptance Test (FAT)." The NRC staff confirmed this to be the case and determined that removal of these test items from Section 7.3.1.3 of the SPM is acceptable because all required test activities will continue to be performed.

Subsection 4.5.2.4 of the SPM discusses metrics used for integration tests.

The Common Q system is an integrated suite of hardware and software designed specifically for nuclear safety applications. Software integration of an application that uses Common Q consists of three components.

1. Integration of software modules to form system executable programs. For a Common Q project this level of integration is accomplished by the creation of control functions using a WEC approved development tool. Proper use of the tool involves assembly of pre-approved Program Control (PC) elements into complete control functions. These control functions are converted into code to be used for transfer to the Common Q hardware. Structured design techniques, including the use of data flow diagrams represent interactions among modular elements and the flow of data among these elements. Unit and Module tests are performed to ensure that the module and system requirements have been met by the integrated software.

   Software used in the flat panel display system (FPDS) is developed in accordance with the SPM processes. FPDS software applications are developed using a WEC approved graphical user interface software tool. Structured design techniques similar to those used for AC160 are also applied to the development processes of the FPDS components. These FPDS applications are then integrated into the FPDS node box and the FPDS hardware is integrated into the application specific Common Q system design.

2. Integration of the resultant programs with the production hardware and instrumentation or with representative functionally equivalent hardware and instrumentation. This level of integration is performed at the manufacturing facility after the cabinets are assembled and energized. Optionally, this integration testing can be performed using surrogate equipment which is functionally equivalent to the production hardware. The system hardware architecture is established in conjunction with the application software; therefore, specific assignment of software programs to PM646A processors is performed prior to the generation of application executable code. The processor applications are loaded into the PM646A processors as the system is prepared for integration testing. An integration test is performed to verify that the released software correctly integrated with the production hardware or representative test bed hardware. All cabinets within a safety system division are interconnected and integrated as a part of the integration test process.

   The NRC staff notes that even in cases where representative equipment is used for integration test purposes, subsequent factory tests must be performed using actual production equipment. Section 7.3.1.5, “Factory Acceptance Test (FAT),” states that “FAT includes tests that are performed on the deliverable system for each deliverable system.”

3. Testing the resulting integrated product. This final level of integration is completed during the System FAT by confirming the correct relationship between test input and output signals. System functions that are implemented across multiple safety divisions are tested to ensure that the overall integrated system meets the systems specifications defined in the System Requirements Specification. For first of a kind systems (FOAKs), certain activities associated with the FAT may have been performed during the system validation tests, and if properly documented, would not need to be re-performed during the FAT. For Nth of a kind systems, the FAT, together with the documentation for prior V&V activities, verifies that all system level functional and performance requirements are satisfied. Regardless of whether the FAT is for a FOAK system or Nth of a kind system,

the purpose of a FAT is to demonstrate the complete system is integrated and functional.

The staff reviewed WEC's application software development and testing processes for both AC160 and FPD software and found they specify how to develop plans for software integration both during the development of the software and during integration with the hardware. The actual integration procedures will be prepared during the planning stage of each project. The staff concludes that the plans for software integration exhibit the management, implementation, and resource characteristics outlined in BTP 7-14 and are, therefore, acceptable.

3.2.5 Software Installation Plan

The acceptance criteria for a Software Installation Plan are contained in BTP 7-14, Section B.3.1.5. IEEE Std. 1074-2006, Clause A.1.2.4, "Plan Installation," endorsed by RG 1.173, provides an acceptable approach for software installation plans. The software installation plan includes the necessary software modifications, checkout in the target environment, and customer acceptance. If a problem arises, it must be identified and reported. BTP 7-14, Section B.3.1.5.4 states that there should be approved procedures for software installation, for combined hardware and software installation, and systems installation. In addition there should be a controlled process to identify, correct, and document errors in the installation procedures.

The Software Installation Plan for Common Q system software is Section 8 of the Common Q SPM. Its purpose is to describe the installation processes to be used for the Common Q system. These processes include loading both operating system and application software into the production Common Q AC160 processor modules and Flat Panel Display system processors.

The staff reviewed the Common Q SPM and found that it included adequate plans for software installation. The procedure(s) for installing the software will be prepared before the installation and checkout phase of the software life cycle. The staff finds that the plans for software installation exhibit the management, implementation, and resource characteristics outlined in BTP 7-14 and are, therefore, acceptable. However, the Common Q Software Installation Plan does not address the installation of the Common Q System into the plant environment. Since the applicant or licensee assumes responsibility, including vendor oversight, for the software installation phase information necessary to address the criteria of BTP 7-14, further evaluation of the site installation activities will be required. This should be accomplished as part of plant specific action item 2.

3.2.6 Software Maintenance Plan

The acceptance criteria for a Software Maintenance Plan are contained in BTP 7-14. Section B.3.1.6. IEEE Std. 7-4.3.2-2003, Clause 5.4.2.3, endorsed by RG 1.152 provides guidance on maintenance and configuration management for commercially dedicated items. IEEE Std. 1074-2006, Clause A.4.2.3, "Maintenance Activity Group," provides an approach for software maintenance plans. IEEE Std. 1074-2006, Clause 6.3.1 states the Maintenance Activity Group is concerned with the identification of enhancements and the resolution of software errors, faults, and failures. NUREG/CR-6101, Section 3.1.9 and Section 4.1.9 also contain guidance on Software Maintenance Plans. These sections identify the maintenance activities to be governed by the Software Maintenance plan as; failure reporting, fault correction, and re-release procedures.

The Software Maintenance Plan for Common Q system software is Section 9 of the Common Q SPM. This plan specifies the requirements for the maintenance and use of Protection class and Important-to-Safety class software used in Common Q Systems. Activities associated with the maintenance phase include:

1. Problem/modification identification, classification and prioritization;
2. Modification analysis;
3. Software maintenance design;
4. Software maintenance implementation;
5. New Software / System test; and
6. Modification delivery.

The staff has reviewed the plan for maintenance of the software as described in the SPM and concludes that it exhibits the characteristics for management, implementation, and resources as set forth in BTP 7-14 and is, therefore, acceptable.

3.2.7 Software Training Plan

The acceptance criteria for a Software Training Plan are contained in BTP 7-14, Section B.3.1.7. IEEE Std. 1074-2006, Clause A.1.2.6, “Plan Training,” endorsed by RG 1.173, provides an acceptable approach to software training plans. If the licensee will be performing the digital system maintenance, the training plan(s) will be more involved, since additional knowledge is necessary to perform maintenance.

Personnel involved in Common Q software design and development are required to have documented training in material covered by the SPM. The requirements for training associated with the Common Q system are addressed within the following sections of the SPM:

- 3.3.3, “Staff Qualifications and Training”
- 3.5.1, “Training”
- 4.14, “Training”
- 7.2.2, “Staffing and Training”

In addition requirements for maintaining Training Materials and Training Records are listed in Table 1, "Document Requirements" and Table 2, “Information Requirements,” for the Common Q system.

The Common Q SPM specifies the requirements for training programs for end users if within Westinghouse’s scope of supply. WEC develops training materials and training programs for use by its Common Q customers. Once delivered, the customer assumes responsibility for providing training to its operators, maintenance and management personnel as appropriate.

All training materials prepared for Common Q customers must be reviewed by the IVV team. For each software system, a separate training program will be developed to ensure safe operation and use of the software within the overall system. The training program will include safety training for the users, operators, and maintenance and management personnel, as appropriate. The SPM stipulates that a training record will be kept on file for each training session, recording the instructor, date, material covered, and personnel attending, to ensure that the appropriate training has been obtained before using the system. The V&V team will review the training documentation for traceability to safety requirements. The training programs

for use at the sites will be developed later. This is an activity that will be influenced by the end users' training facilities and procedures. The staff concludes that the specified plans for training of the software developers and end users meet the criteria outlined in BTP 7-14 and are, therefore, acceptable.

3.2.8 Software Operations Plan

The acceptance criteria for a Software Operations Plan are contained in BTP 7-14, Section B.3.1.8. IEEE Std.1074-2006, Clause A.4.2, endorsed by RG 1.173, provides guidance for software operations plans. IEEE Std.1074-2006, Clause A.4.2 states an operation and support process involves user operation of the system and ongoing support. Support includes providing technical assistance, consulting with the user, and recording user support requests by maintaining a Support Request Log. Thus, the Operation and Support Process may trigger Maintenance Activities, which the Software Maintenance Plan should address. IEEE Std.1074-2006, Clause A.4.2.1.2 states that the Installed Software System shall be utilized in the intended environment and in accordance with the operating instructions.

The revised version of the SPM, does not contain a dedicated section to address the criteria for software operations planning. WEC stated that the Software Operations Plan is either a project specific activity or the Licensee's responsibility.

The Software Operations Plan is not within the scope of the Common Q Software Program Manual. Therefore, a safety determination cannot be made for a Software Operations Plan in this regard. Since the applicant or licensee will assume responsibility, including vendor oversight, for the software operations phase of the software life cycle, relevant information must be evaluated as part of a plant specific action item. An evaluation of compliance with the criteria of BTP 7-14 Section B.3.1.8 shall be performed at the time of system development when the operational aspects of the system have been defined. These requirements are captured as PSAI's 3 and 4.

3.2.9 Software Safety Plan

BTP 7-14, Section B.3.1.9 provides guidance to evaluate software safety plans (SSP). The SSP should require that appropriate safety requirements be included in the software requirements specification. The SSP 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 describe the boundaries and interfaces between the software safety organization and others. It should show how the software safety activities are coordinated with the development activities and the interactions between software safety organization and the software V&V organization. SSP should designate a single safety officer who has clear responsibility for the safety qualities and has clear authority to accomplish the goals of the safety requirements in the SRS design, and implementation of the software.

The Software Safety Plan for Common Q system software is Section 3, Software Safety Plan, of the Common Q Software Program Manual. The stated purpose of the Common Q Software Safety Plan is, *"…to enable the development of safety critical software for Common Q™ Systems that has reasonable assurance that software defects do not present severe consequences to public health and safety."*

To accomplish this goal, the Common Q SSP defines procedures and methods to be used for the development, procurement, maintenance and ultimately, retirement of all protection class

Common Q software. The other classes of Common Q software; Important to Safety, Important to Availability, and General Purpose, are not included in the SSP because they are not considered to be safety critical. This is because the failure of this software would not result in severe consequences to public health and safety.

Software Safety Organization:

The Common Q SSP establishes a software safety organization which is composed of two parts. The first part is the quality organization, which is an independent quality assurance department. This quality organization coordinates and reviews quality assurance procedures and directives. The Quality organization has a reporting chain separate from the design team such that the QA organization is independent of project schedule and cost considerations. The Quality organization provides oversight by way of periodic audits to verify that the Automation Engineering organization is correctly abiding by both the procedures and directives generated by both organizations. The SSP is approved by the Manager of the Quality organization, or designee.

The second part of the software safety organization is the Independent Verification and Validation Team (IVV Team). This IVV team performs the safety activities for a given Common Q system implementation project.

The resource requirements needed to perform software safety activities are to be developed by the IVV team leader and the Engineering Project Manager. A plant specific Project Quality Plan will coordinate both the system development, software safety and quality assurance activities to identify the prescribed procedures and provide the resources needed for their execution.

During the requirements phase of the software development life cycle process, an evaluation is performed to identify the safety critical hazards posed by the system through its interfaces. For each hazard identified, the analysis determines whether a software malfunction could produce the hazardous condition. Each software producible hazard is then subsequently evaluated during each development phase of the safety critical software to determine if new hazards have been introduced during that phase, or if the evolving design has altered the results of the hazards analysis. The results of IVV analyses performed on requirements, design, code, test and other technical documentation are documented in the IVV Phase Summary Reports and the Final IVV Report for the system.

The safety requirements that need to be met by the software in order to mitigate or control system hazards are defined in the system requirements specifications. The software design description will include descriptions of the software design elements that satisfy the software safety requirements. The responsibilities for the execution of the SSP and for ensuring that the software safety activities are completed in accordance with the plan are divided between the IVV Engineering Line Manager (ELM) and the quality manager.

The safety organization defined in the Common Q SSP considers the security risk as well as the risk to the plant if the digital system malfunctions. The critical design review identifies the risks associated with the system design in a manner that is consistent with the software safety strategy.

The staff has reviewed the Common Q SSP and finds that it addresses the topics described in the SRP and in IEEE Std. 1228-1994 (Reaffirmed in 2002), “IEEE Standard for Software Safety Plans.” The Common Q SSP describes the organizational structure and responsibilities,

resources, methods of accomplishment, and integration of system safety with other program engineering and management activities. The hazards evaluations required by the SSP will be documented in the V&V documentation. The Common Q SSP identifies the international, national, industry and company standards and guidelines to be followed by the safety organization. The staff determined the software safety activities defined in the SSP will adequately identify and resolve safety issues associated with the Common Q software. The staff concludes that the Common Q SSP adequately addresses the topics outlined in the SRP and is, therefore, acceptable.

3.2.10 Software Verification and Validation Plan

The acceptance criteria for the SVVP are contained in the SRP, BTP 7-14, Section B.3.1.10, "Software Verification and Validation Plan," and Section B.3.2.2, "Acceptance Criteria for Software Verification and Validation Activities." These sections identify RG 1.168, "Verification, Validation, Reviews, and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants" which endorses IEEE Std. 1012-2004, "IEEE Standard for Software Verification and Validation," as providing methods acceptable to the NRC staff for meeting the regulatory requirements for verification and validation of safety system software. This section also states that further guidance can be found in NUREG/CR-6101, Sections 3.1.4 and 4.1.4.

Verification is defined as the process of determining whether the products of a given phase of the development cycle fulfill the requirements established during the previous phase. Validation is defined as the test and evaluation of the integrated computer system to ensure compliance with the functional, performance, and interface requirements.

Combined, verification and validation is the process of determining whether the requirements for a system or component are complete and correct, the products of each development phase fulfill (i.e., implement) the requirements to meet the criteria imposed by the previous phase, and the final system or component complies with specified requirements.

The Software V&V Plan for Common Q system software is Section 5 of the Common Q Software Program Manual. The stated purpose of the Common Q SVVP is to establish requirements for the IVV process to be applied to Common Q systems. It also defines when, how and by whom specific IVV activities are to be performed.

The aim of the Common Q software V&V program is to provide an acceptable generic methodology of V&V as part of the qualification process for computer software applications developed for the Common Q platform. The Common Q SVVP applies to all new software to be developed under the SPM and to some previously developed application software to be used in the Common Q platform. For the qualification of existing software, either for use in the generic Common Q platform or for use in new applications, the following cases are identified:

- Existing commercial software will be qualified under the Commercial Grade Dedication Program, which is outlined in the Common Qualified Platform Topical Report (Ref. 14).

- Existing non-commercial software that has been actively used in nuclear power plants will be qualified for the Common Q platform by judging its original V&V program. The V&V effort will make this judgment using review criteria similar to those for newly developed software.

Other existing non-commercial software may be used under the conditions that (1) the software fulfills a specific requirement identified in the software requirements specification, (2) the code is well organized and has adequate design documentation and source code commentary to permit the application of the V&V process, and (3) the software is subjected to the V&V process, starting at the design phase.

For the development of new application software, depending on the scope of each specific project, WEC will decide whether to issue a project-specific SVVP or to maintain the generic plan as is. The use of the generic plan will require that the software developers manage the deviations and the project-specific aspects through the project-specific plan to be developed for each project. WEC will hold these project-specific SVVPs for audit. WEC also will hold the project-specific V&V reports for projects developed under the Common Q platform for audit, and the licensees will hold the V&V reports associated with plant-specific applications for audit. Succeeding systems manufactured under the same design as a system that was previously verified and validated in accordance with this SVVP will be certified by performing, as a minimum, the equivalent of the validation tests that were applied to the verified and validated system. The staff considers this approach to be acceptable.

WEC differentiates the span of the V&V activities and the grade of independence required for V&V reviewers according to the classification of each software item. The Common Q software integrity level classifications have been updated in Revision 5 of the SPM and are discussed in Section 3.2.2 of this SE. These Classifications are:

- Protection,
- Important to safety,
- Important to availability, and
- General purpose.

These four levels respectively are matched to the four categories in IEEE Std. 1012-2004 of 4, 3, 2, and 1. The Software Integrity Levels described in the Common Q SPM are mapped to the activities associated with IEEE Std. 1012-2004, SIL 4 in the SPM.

WEC follows the guidance provided in IEEE Std.1012-2004 regarding structure and content for SVVPs when applied to the development of safety-related Common Q software. IEEE Std. 1012-2004 provides the uniform and minimum requirements for the format and content of these plans. Additionally, the standard defines the minimum set of specific V&V tasks to be carried out during each phase of the critical software development life cycle and the required inputs and outputs for these tasks. Exhibit 5-8 of the SPM lists and defines the specific V&V tasks used for Common Q software development and maps these tasks to the V&V activities defined within IEEE Std. 1012-2004. The tables in Exhibit 5-1 and 5-8 identify the minimum set of V&V activities for all classifications of Common Q software including noncritical software. The NRC notes that V&V Tasks for Important to Availability and General Purpose classifications are identified in Exhibit 5-1.

The Common Q SVVP incorporates verification reviews and validation testing. Verification reviews are supported by the use of checklists and requirements traceability analyses for the phases of requirements, design, implementation, test, and installation and checkout. A requirements traceability matrix will be prepared at the beginning of the software development process and updated throughout the phases of the software life cycle.

Validation testing includes structural and functional testing. Structural testing is performed on software modules and units by path testing. Module and unit testing will be performed in accordance with IEEE Std.1008-1987, "IEEE Standard for Software Unit Testing" (endorsed by RG 1.171). Functional testing is performed on the integrated computer system to determine whether the system meets its functional requirements (functional operations, system level performance, external and internal interfaces, stress testing, testability, and other requirements, as stated during the concept phase).

For protection and important to safety software, verification reviews are performed by the V&V staff. V&V activities for the preparation of test plans, procedures, test result reports and execution of tests are performed by either the design team or by the V&V team depending on the classification level of the software being tested. Exhibit 5-1 of the SPM designates which team is responsible for performing these activities. When the design team prepares the material or executes the tests, the V&V team will oversee the conduct of these activities by reviewing documentation and witnessing testing.

Revision 5 of the SPM introduces a System Validation Test process to validate the hardware design, software design, and system integration of first instance applications at a functional level. Section 7.3.1.4, "System Validation Test," of the SPM was added to the SPM to describe the System Validation Testing activities. Section 7.3.1.5 "Factory Acceptance Test," of the SPM has also been rewritten to adopt the new System Validation Test processes and to describe differences between validation activities performed during Factory Acceptance Testing and validation activities to be performed during the new System Validation Test activities. Exhibit 7-1 in the SPM provides a comparison of System Validation Test and Factory Test Processes.

Validation Test requirements are accomplished for each Common Q system through a combination of System Validation Test activities and Factory Acceptance Test activities.

The System Validation process is intended to be used to validate the first application or first instance of a system design while subsequent instances of the same design will undergo integration testing during Factory Acceptance Test processes. Factory Acceptance Tests will be limited in scope such that testing of logic that was previously verified during System Validation Testing will not be performed. For example, Factory Acceptance Testing will only include a subset of voting logic combinations to demonstrate each input to voting logic is effective whereas System Validation Testing of Voting Logic includes testing of all combinations including bypasses and forced trips.

System Validation Testing can also be performed using representative Common Q equipment in lieu of production hardware to be delivered and installed into a licensed facility. Conversely, Factory Acceptance Tests are performed on the deliverable system, both the hardware and software, and are performed for each deliverable Common Q system.

Test documentation will be prepared in accordance with IEEE Std.829-1998, "IEEE Standard for Software Test Documentation." IEEE Std. 829-1983 is endorsed by RG 1.170, September 1997. After the system is validated, a Code certificate is issued certifying that the system is acceptable for use. The SVVP addresses V&V activities associated with the operation and maintenance phase by ensuring that program modifications are submitted to the same V&V program applied to new software development. Software changes will be evaluated by a software safety change analysis, the results of which shall be found in the V&V report. The SVVP addresses the use of regression testing for the V&V of software modifications.

The SVVP also addresses activities designed to verify the adequacy of the software development documentation issued throughout the software life cycle, installation procedures, training materials, and user documentation.

As a result of the V&V activities throughout the software development process, V&V phase summary reports, including discrepancy reports, will be issued. A final V&V report will be issued after the V&V process, including the assessment of the overall software and system quality and a Code certificate. Results of V&V analyses performed on requirements, design, code, test, and other technical documentation are documented in the V&V phase summary reports and the final V&V report. Information on suspected or confirmed safety problems in the pre-released or installed system is recorded in the final V&V report. Results of audits performed on software safety program tasks are documented in the V&V phase summary reports and in the final V&V report. Results of safety tests conducted on all or any part of the entire system are documented in the test report. Software safety certification is documented in the Code certificate. The SVVP is reviewed for adequacy and completeness of the V&V methods by an independent reviewer.

The staff has reviewed the information in the SVVP regarding software module testing and concludes that the procedures used for performance of software module testing satisfy the software V&V program requirements of IEEE Std. 7-4.3.2-2003 and are, therefore, acceptable.

Independence of Verification and Validation

The independence requirements for organizations performing quality control activities are addressed by 10 CFR Part 50 through Criterion I and Criterion III of Appendix B. Criterion I requires in part, that individuals and organizations performing quality assurance functions have sufficient authority, organizational freedom and independence from cost and schedule. Criterion III requires that individuals or groups performing design control activities be different from those who performed the original design, but they may be from the same organization.

The positions reflected in specific standards addressing V&V activities associated with the implementation of digital I&C systems vary from requiring only technical independence, as in RG 1.152 by endorsing IEEE Std.7-4.3.2-2003, to requiring technical, financial and schedule independence, as in RG 1.168. IEEE Std.1012-2004, endorsed by RG 1.168, does not specifically address the level of independence required. IEEE Std.1012-2004 includes an informative annex contemplating the position that for high-integrity-level software, the level of independence required for the V&V organization encompasses technical, managerial, and financial independence.

The organization responsible for ensuring that the Common Q software has been developed according to the quality required by its classification (called the software safety organization in the SPM) is composed of two parts:

- An independent quality assurance organization, which performs the verification of the implementation of quality assurance requirements according to Appendix B of 10 CFR Part 50. This organization, outside the cognizant engineering organization (CEO), generates the quality assurance procedures and directives that are followed by all CEOs.
- An independent V&V Team within the CEO that performs the safety activities of the CEO for a given Common Q system implementation project.

Within the CEO, software activities are organized into two teams: the design team, responsible for the development of the software, and the V&V Team, which performs the testing of the system as well as the V&V activities. The director of the CEO is responsible and accountable for both technical and administrative aspects associated with the development and V&V tasks for each system assigned to the CEO. The director or manager may assign a project manager to be responsible for the development of the software for a specific Common Q project. The CEO Director assigns the appropriate resources to the project manager and the V&V team leader. Members of the V&V team are not allowed to participate on the design team, even on a part-time basis, while a safety-class system is being designed. The V&V team leader, responsible for the V&V, must not be the design team leader. Additionally, the independent reviewer must also be competent to perform the review.

In response to RAI 11 (Refs. 15 and 16), Westinghouse provided clarification of IVV group membership. The SPM further states that; "The IV&V Team in the context of this SPM refers to those individuals within the IV&V organization who perform V&V functions on the safety system design, implementation, and test (i.e., engineers and technicians). The IV&V organization may include other individuals who perform supporting roles that are not design verification related and the organizational independence does not apply to those individuals."

The SPM states that the V&V leader is responsible for the schedule and budget for the V&V activities, the project manager is responsible for the schedule and budget for the activities associated with the software development and, therefore, financial and managerial independence between the development group and the V&V group is achieved.

The staff finds that the WEC approach on independence of V&V for the Common Q platform is in accordance with the requirements of IEEE Std.7-4.3.2-2003, and is compatible with IEEE Std. 1012-2004, "IEEE Standard for Software Verification and Validation," as endorsed by RG 1.168 and is, therefore, acceptable.

3.2.11 Software Configuration Management Plan

BTP 7-14, Section B.3.1.11 provides guidance for the evaluation of the Software Configuration Management Plan, and states that IEEE Std.1074-2006, Clause A.1.2.2, "Plan Configuration Management," provides an acceptable approach to software configuration management. IEEE Std.1074-2006, Clause A.2.2.2.2 states that Software configuration management includes the evaluation, coordination, approval or disapproval, and implementation of changes to product components (e.g., code, documentation) after a baseline has been established. Items that are to be managed should include code, documentation, plans, specifications, project policies, procedures, and other artifacts. BTP 7-14, Section B.3.1.11.1 calls for the definition of the responsibilities and authority of the Software Configuration Management (CM) organization.

The Software Configuration Management Plan (SCMP) for Common Q system software is Section 6 of the Common Q SPM. The SCMP is applicable to all Common Q software as well as software tools used in the development of Common Q software. The Common Q SCMP describes the organizational structure that controls the configuration of software. Software Configuration Management is intended to be applied throughout the entire software life cycle, including requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and retirement phase.

The design team and the IVV Group in the Nuclear Automation organization are responsible for implementation of adequate measures to manage and control the software configuration of a

Common Q project. The Common Q SCMP describes the independence of those responsible for system software configuration management functions from those responsible for verification and validation activities related to configuration management. The SCMP describes the process for configuration control including configuration identification, software change request, software change authorization, module and unit release history, baselines, and backups. The SCMP describes the software configuration management activities related to the software project baselines, the configuration change control authority and management, methods of access control, and the configuration status control log maintenance. Project-specific configuration management data that reflect the specific methods of managing the software configurations will be developed as part of the project plan required for every Common Q project. The SCMP identifies the international, national, industry, and company standards and guidelines to be followed for the software configuration management activity.

The staff concludes the SCMP conforms to the requirements identified in IEEE Std. 828-2005, which is endorsed by RG 1.169. This meets the criteria of BTP 7-14 and is, therefore, acceptable.

3.2.12 Software Test Plan

The acceptance criterion for STP is contained in the SRP, BTP 7-14, Section B.3.1.12, "Software Test Plan," and in Section B.3.2.4, "Acceptance Criteria for Testing Activities." These sections state that both RG 1.170, September 1997, "Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants," which endorses IEEE Std. 829-1983, "IEEE Standard for Software Test Documentation," and RG 1.171, "Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants," which endorses IEEE Std. 1008-1987, "IEEE Standard for Software Unit Testing," identify acceptable methods to satisfy software unit testing requirements.

The Software Test Plan (STP) for Common Q system software is Section 7 of the Common Q Software Program Manual. This plan identifies the testing activities and test documentation required to verify and validate Common Q safety system software. The scope of the STP includes testing of Common Q platform component software as well as application software that is developed with the Common Q platform.

The Common Q STP describes and defines the test activities for the following test types:

- Module Tests
- Unit Tests
- Integration Tests
- System Validation Tests
- Factory Acceptance Tests

The Module level tests are performed to confirm proper functionality of the platform level software components of Common Q. These tests are not application specific and are used to develop a library of approved building blocks to be used for application development.

Unit tests are performed during the plant specific system design to ensure proper functionality of the platform components as they are incorporated into a specific application.

Integration tests are used to confirm that the program units have been properly connected and are integrated in a manner to ensure proper operation of the overall system. Integration tests are conducted on the target hardware to be installed at the plant site so they also confirm the proper integration of software to the hardware of the system.

Validation Test requirements are accomplished for each Common Q system through a combination of System Validation Test activities and Factory Acceptance Test activities.

System Validation Tests are performed to validate the hardware design, software design, and system integration of first instance applications at a functional level. The System Validation process is intended to be used to validate the first application or first instance of a system design while subsequent instances of the same design will undergo integration testing during Factory Acceptance Test processes. System Validation Testing can be performed using representative Common Q equipment in lieu of production hardware.

Factory Acceptance Testing of the system is conducted with the final application software installed on the targeted hardware that has been assembled.

Revision 5 of the SPM adds a provision that allows FAT some activities to be performed after system delivery to the site. This position was clarified in Westinghouse's response to RAI 5 (Refs. 15 and 16). The revised Section 7.3.1.5 of the SPM states: *"The FAT is typically performed in the factory but some portion of the test can be performed at site if agreed to with the customer."* The FAT objectives include demonstration that the complete system is integrated and functional. The NRC staff determined this change is acceptable because the objectives of the FAT as stated in Section 7.3.1.5 of the SPM will continue to be accomplished prior to the system being placed into service even if some FAT activities are deferred to the site.

The FAT is the final stage of testing that is conducted prior to acceptance of equipment by the licensee. All subsequent testing activities such as Site Acceptance Testing and Installation testing are considered to be the responsibility of the licensee and are therefore not within the scope of the Common Q STP. The Common Q STP identifies the following two categories of testing that are used in the Common Q software testing process;

- Functional Testing - (otherwise known as black box testing) is used to determine that a module or system has functional performance that is consistent with the requirements specified. Test cases for functional testing are derived from the requirement specifications and are based on manipulating test inputs and monitoring test outputs.

- Structural Testing - (otherwise known as white box testing) is used to evaluate the internal structure of a code module and is only used for module tests. Structural testing is intended to provide one hundred percent of branch execution within the code module.

Section 7.2.4 of the SPM Revision 5 includes new provisions for deferring completion of test activities to allow commencement of the subsequent tests before the preceding test level is complete. This change was further clarified in Westinghouse's response to RAI 4 (Refs. 15 and 16). This change is being made to account for the fact that modules can either be generically produced (existing software not to be modified during application development) or may be specifically developed or modified for a particular project (new software, or existing software to be modified during application development).

When pre-validated modules are used for an application, the project's validation testing can begin with Unit Testing of the released application. When specifically developed modules are used, validation of the software module (module test) can be performed while the application software that uses the module is concurrently undergoing downstream validation tests. Westinghouse recognizes this is a calculated risk in project validation testing, and should the module test fail while downstream testing is occurring concurrently, the downstream validation testing may be required to be reperformed to demonstrate valid downstream testing results.

The NRC staff determined this change is acceptable because all testing requirements for each level of test will continue to be met even though the test sequence can, in some cases, be changed to support application specific requirements.

The risks associated with software testing are addressed through regression analysis. The STP states that *"regression analysis shall be performed to determine extent of retesting activities that may be necessary to re-verify and/or re-validate any changes to a tested element."* The results of this analysis are intended to identify latent design errors or programming bugs that have been introduced by software design modifications.

The Common Q STP prescribes the scope, approach, resources, and schedule of the testing activities and it identifies the items and features to be tested. Testing tasks as well as the personnel responsible for each task are identified. The software test plan includes module testing, unit testing, integration testing, System Validation Testing and factory acceptance testing.

Revision 5 of the SPM removes the requirement for test plans to contain all the requirements for all acceptance test procedures and to define each required test to be conducted. The reason for this change was provided by Westinghouse in response to RAI 12 (Refs. 15 and 16). This response states the following:

> The reason for the change in the SPM is due to the typical sequence and progression of a project. Requirements analysis, testing coverage and tracing of the requirements to test cases are significant testing activities. The Test Plan is needed to outline these activities. The test planning and initial engineering work occurs in parallel with the finalization of the design requirements and the implementation specifications. Therefore, the specific requirements to be tested are not available or issued in their final form when the test plan is written.

The NRC staff determined this change to be acceptable because Westinghouse's processes will continue to establish traceability between system requirements and test procedures and/or test cases even if these are determined after the test plan is written. As such, the individual requirements for lower level acceptance test procedures and identification of individual, specific required tests to be conducted do not need to be included in the test plan itself at the requirements phase of development and can instead be established at a later stage of the development process.

Site acceptance testing and installation testing are not covered under the Common Q STP because they are considered to be licensee actions and are to be addressed during the development of a Common Q based application. As such, a project specific test plan should be developed and used to address these aspects of software test planning. This is addressed in plant specific action item 5.

The Common Q STP is understandable and it includes adequate provisions for retest in the event of failure of the original test. The Common Q Software Test Plan adequately addresses the test planning guidance of BTP 7-14, Section B.3.1.12, and based on WEC's commitment to conformance with IEEE Std. 829-1998 and IEEE Std.1008-1987, the staff finds the Common Q Software Test Plan acceptable.

3.2.13 Secure Development and Operating Environment (SDOE) Evaluation

The staff evaluated the Common Q platform requirements against RG 1.152. It contains five regulatory positions that describe methods acceptable to the staff for establishing an SDOE for digital safety systems. Each of these positions correlates to a phase of a typical software development life cycle. These regulatory positions support compliance with portions of 10 CFR Part 50 – specifically Appendix A GDC 21 (Protection System Reliability and Testability), Appendix B Criterion III (Design Control) and IEEE Std. 603-1991, Clauses 5.6.3 (Independence from Interconnected Equipment) and 5.9 (Access Control).

Section 12 of the Common Q Software Program Manual (Ref. 14) addresses the SDOE planning aspects of the Common Q platform from the Concepts Phase through the Test Phase of the software development life cycle per the guidance provided by RG 1.152. In addition, an applicant or licensee using a Common Q platform based system must perform actions to satisfy PSAI 7.

The lifecycle structure, for which criteria on development environment controls are to be established, consists of the following phases:

- Concept
- Requirements
- Design
- Implementation
- Test
- Installation, Checkout, and Acceptance Testing
- Operation
- Maintenance
- Retirement

This SE evaluates the secure development environment controls applied to the Common Q safety system development from concept phase through the test phase. The last four phases: Installation, Operation, Maintenance, and Retirement will need to be evaluated via follow-up activities once a safety system application is developed using the Common Q platform.

The operating software for the Common Q platform was developed prior to the issuance of RG 1.152. Thus the discussion of development activities is focused on those secure development environment considerations applied during the commercial grade dedication effort applicable to the life cycle processes for maintenance of the previously developed software. Although application software is not within the scope of this review, platform features that contribute to the SDOE for the application are identified and discussed. Credit may be taken for the use of these security capabilities in establishing a secure operational environment for a plant specific safety-related application.

A security evaluation for the Common Q platform was not conducted by the NRC when the Common Q platform SE (Ref. 2) was performed because the applicable regulatory guidance was not available at the time of that safety evaluation. Nonetheless, the security measures discussed below were in place during the Common Q platform development.

3.2.13.1 Concepts Phase (2.1)

Secure Operational Environment Capabilities

The Common Q platform was developed prior to the issuance of regulatory guidance on security capabilities. The security enabling capabilities of the Common Q platform were not implemented to fulfill a specific security concept, but were rather the product of good design practices. The NRC staff review of the Common Q development documentation determined that the development process incorporated several security features in the original design that apply to the secure development and operating environment of the system. Even though a formal concepts phase security analysis was not performed, the WEC SDOE plan supports the security concepts used during the development of the Common Q platform. The basic concepts used in defining the system security capabilities of the Common Q platform were ensuring confidentiality, and integrity. The vulnerabilities associated with these concepts are defined in the SPM as follows.

- *Confidentiality Vulnerability - the inadvertent loss of information related to the security of a system and related development systems.*

- *Integrity Vulnerability - the inadvertent change to a system and related development system design requirements that could adversely affect security*

The security capabilities of the Common Q platform that include physical and logical access controls, safety to non-safety isolation, and control of the various life cycle activities, were derived from these security concepts. These security capabilities were used to establish the security requirements for the system hardware and software. Even though the Common Q platform was developed several years prior to the issuance cyber security regulatory guidance, the NRC staff review concludes that the WEC SDOE plan satisfies the criterion for identifying safety system security capabilities.

General Life Cycle Vulnerabilities

A formal security assessment for the Common Q platform design was not performed at the time of development because the platform was designed prior to the availability of guidance in this area. Instead, WEC provided a SDOE plan which includes an analysis of the vulnerabilities applicable to the development of the Common Q platform. This is an acceptable alternative approach considering the fact that the Common Q platform design was completed prior to the issuance of RG 1.152.

The SPM calls for V&V activities to be performed during the Concept, Requirements, Design, Implementation, and Test phases to verify correct implementation of secure operational environment requirements.

The vulnerabilities of the Common Q platform development are initially assessed during the concepts phase. Subsequent assessments are also performed to determine if new vulnerabilities are introduced to the system during the later stages of the development process. The NRC staff finds that these identified vulnerabilities and the applicants response to them

adequately address the potential for tampering with the Common Q platform during its developmental phases. The vulnerabilities identified by WEC were used to derive the security controls for the system hardware and software development. Based on the review of identified vulnerabilities and the fact that requirements to address these vulnerabilities through the various life cycle phases are described in the SDOE plan, the staff has determined that the Common Q SPM adequately identifies and addresses the vulnerabilities associated with software development.

Remote Access and One-Way Communication

The Software Program Manual states that Isolated Development Infrastructures (IDI) are created to preclude inadvertent and remote access or changes that could affect the confidentiality or integrity of a system and related development system hardware or software during the implementation phase. The NRC staff understands this to mean that Common Q systems under development will be configured in an isolated manner which precludes any remote access to the safety system. Though the Common Q system can be configured to provide remote access capability, measures are taken by the design and development team to prevent the implementation of these features. WNA-DS-01070-GEN-P Rev. 6, "Westinghouse Application Restrictions for Generic Common Q," (Reference 5) is used to identify generic restrictions that are applied to all Common Q projects. This document identifies several measures that are taken to prevent remote access to the PM646A safety processors including a measure to prevent software installation over the AF-100 bus, as well as a measure to restrict network connectivity of the serial interfaces on the processor module. An additional requirement to disable the remote access capabilities in the application is also described. The NRC staff determined that the Common Q SPM provides adequate provisions to establish one way communications where required and to prevent remote access to the safety system.

The staff finds that the Common Q SDOE plan adequately addresses the criteria of position C.2.2.1 of RG 1.152.

3.2.13.2 Requirements Phase (2.2)

System Features (2.2.1)

Security functional performance requirements are implemented to address vulnerabilities identified in the concept phase for the Common Q system. All such requirements are subject to independent verification and validation as part of the overall IVV process.

NRC staff finds that the requirements pertaining to the security functions, system configuration, external interfaces, qualification, human factors, data definitions, and documentation for hardware and software have been properly established and are therefore acceptable.

The Common Q SPM has provisions for a security assessment to be performed during the concept phase. The results of the security assessment are security related design features. Security related design features are implemented into the system requirements specifications. The Common Q SVVP states that the IVV team evaluates the software design and test documentation, which includes the system requirements specification. As such, the system requirements specification which includes security related design features is evaluated by the IVV team.

NRC staff finds that the verification process used for security related design features provides an adequate means of ensuring the correctness, completeness, accuracy, testability, and consistency of the system's security features and is therefore acceptable.

Previously Developed Common Q software

The previously developed operating software of the Common Q platform is dedicated for use in safety-related applications. As described in Section 4.2 of the Common Q platform Topical Report SE (Refs. 1 and 14), commercial-grade dedication is an acceptance process for demonstrating that a commercial grade item to be used as a basic component will perform its intended safety functions and, in this respect, is equivalent to an item designed and manufactured under a 10 CFR Part 50, Appendix B quality assurance program. Testing performed as part of the commercial grade dedication effort further establishes the quality and security characteristics of the previously developed software. The dedicated operating software is controlled under the Common Q software configuration management program (SCMP) as evaluated in Section 3.2.11 of this SE and is maintained under the Common Q Quality Assurance program which is evaluated in Section 3.2.3 (SQAP) of this SE. Based on the review of the evidence for the previously developed software and its ongoing management under the WEC quality processes, the NRC staff determined that the Common Q previously developed software satisfies the criterion of regulatory position C.2.2.1 in RG 1.152.

Development Activities (2.2.2)

Among the identified vulnerabilities of the Common Q system was its vulnerability to inadvertent change to the design requirements of a system or related development system that could adversely affect the security of the system. If appropriate controls are not placed within the requirements development process, then the opportunity exists for inappropriate requirements to be inserted and/or necessary requirements to be omitted. The actions taken by WEC to prevent requirements tampering are described below.

During development of the Common Q platform software, the SPM defines configuration management, quality assurance, and life cycle development processes used to control activities performed in the requirements phase. The engineering procedures used by WEC govern the organization, content and structure of requirements specifications for the Common Q platform.

The software review process, including responsibilities, review methods, review processes, and specific review activities are defined in the Common Q SQAP. The Reviews section of the SPM (Section 4.6) addresses the review requirements throughout the software life cycle. A Software Requirements Review (SRR) is required to be performed by the IVV team after the completion of the requirements phase. During this SRR, an examination of the software requirements specifications is performed to verify that they are clear, verifiable, consistent, modifiable, traceable and usable during the operations and maintenance phases. The SRR includes an evaluation of the traceability and completeness of the requirements as well as the adequacy of rationale for derived requirements. The NRC staff review of the Common Q review processes found them to be acceptable and compatible with IEEE Std. 1028-2008 "IEEE Standard for Software Reviews."

The staff finds the measures identified in the Common Q SDOE Plan (Section 12 of the SPM) adequate to prevent inadvertent, unintended, or unauthorized modifications to the system during the requirements phase. The staff also finds the verification activities completed by the IVV team, to be sufficient to identify and mitigate any unauthorized modifications of the Common Q

platform requirements specifications. The Common Q SDOE Plan therefore satisfies the requirements of regulatory position C.2.2.2 in RG 1.152.

3.2.13.3 Design Phase (2.3)

The Common Q system development process has provisions for the creation of a Software Design Description (SDD) which includes descriptions of the software design elements that are used to satisfy software safety and security requirements. The documentation requirements for the SDD are provided in SPM Section 10.3. Here it is stated that "*the SDD … complies with the system requirements specification and the software requirements specification"*. All design features including those that are security related are described in the SDD.

Verification

Section 10.3 of the SPM states that; *"…each software safety design element identified that satisfy the software safety requirements, such that its achievement is capable of being verified and validated per the SVVP."* Therefore, the security design elements of the SDD will be subject to a formal verification and validation process. The evaluation of the Common Q SVVP is documented in Section 3.2.10 of this SE. The staff finds the verification activities completed by the IVV team during the design phase to be sufficient to identify and mitigate any unauthorized modifications of the Common Q platform design products.

Access Controls

Control over the use of safety system services is addressed by the Development System Requirements. These include physical and logical access controls to Common Q system functions. Control of data communication between the Common Q safety system and other systems has been evaluated in Section 4.1.3.4 of the Common Q Platform Topical Report SE (Refs. 1 and 14).

Common Q physical and logical access features are included in the development system requirements and were derived from the vulnerability assessments performed starting in the concept phase of software development. The staff finds this approach to establishing physical and logical access controls for the Common Q system to be acceptable.

Software Configuration Management

The Common Q SCMP defines the process used for identifying software configuration items. During the requirements phase, the Design team and the IVV group perform the tasks of:

- identifying software items developed under SPM for generic application that are to be controlled via the SCMP,
- assuring that the qualification of these items are complete and appropriate for the project (including appropriateness of software classification), and
- describing how the software will be integrated with the project-specific software development.

During the design phase, the system security requirements are translated into these design configuration items. The secure operational environment requirements for the Common Q platform correspond to security-related features, capabilities, and design elements that serve as design configuration items. The staff finds that the process employed for Common Q systems to transfer security functional performance requirements into system design elements is

acceptable. The staff has therefore determined that the Common Q SDOE Plan satisfies the requirements of regulatory position C.2.3.1 in RG 1.152.

Development Activities (2.3.2)

The security measures implemented in the design phase included; system features, verification, access controls, and software configuration management. The staff finds the measures identified in the Common Q SDOE plan adequate to prevent inadvertent, unintended, or unauthorized modifications to the system during the design phase to address Regulatory Position C.2.3.2 of RG 1.152.

3.2.13.4 Implementation Phase (2.4)

Module coding is performed and existing qualified software is integrated into the software system during the Implementation phase of the Common Q software development process. The IVV team also reviews the design team's implementation products during this phase. The SPM states that *"The purpose of the implementation verification is to ascertain the implementation documents are clear, understandable, logically correct and a faithful translation of the design specifications."* It also states that *"The objectives of the implementation documents are to facilitate the effective production, testing, use, transfer, conversion to a different environment, future modifications, and traceability to design specifications."*

System Features (2.4.1)

The V&V activities to be performed during the implementation phase include performing a security assessment of the system to verify that the security controls chosen in the design phase have been properly implemented. If system vulnerabilities are identified during this security assessment then requirements for additional security controls can be added to the system requirements to address or otherwise mitigate these vulnerabilities.

These V&V activities defined in the SPM provide a means by which the correctness and accuracy of the design configuration items produced during the implementation phase can be confirmed. The Common Q development process also includes a process for establishing and maintaining requirements traceability as is described in Section 5.4.5.3 of the SPM. This process involves associating requirements with documentation and software design configuration items. During the requirements traceability analyses that are performed throughout the development process, assessments of completeness are made in order to ensure that; a) all system requirements are implemented and that b) no features are implemented within the design that are not associated with an approved specification.

The NRC staff has reviewed the implementation controls outlined in the SPM and has determined that the Common Q platform development process contains features that comply with the criterion in Section 2.4.1 of RG 1.152.

Development Activities for the Implementation Phase (2.4.2)

The secure development environment established during development of the Common Q system software involves creation of Isolated Development Infrastructures (IDI). These IDI's are intended to preclude inadvertent and remote access or changes that could affect the confidentiality or integrity of a system and related development system hardware or software during the implementation phase.

The SPM establishes requirements for security procedures and standards to minimize and mitigate tampering with the developed system. The security program established by these procedures addresses hidden functions and vulnerable features embedded in the code. Where possible, the program requires these functions to be disabled, removed, or addressed to prevent any unauthorized access.

Use of Commercial-Off-the-Shelf Systems (COTS)

The security program established by the Common Q SPM includes assessments of COTS systems to confirm that the features within the COTS system do not compromise the security requirements of the integrated Common Q system. Additionally, these assessments ensure that security functions are not compromised by the other system functions.

The NRC staff determined that the criterion of regulatory position C.2.4.2 of RG 1.152 has been met.

3.2.13.5 Test Phase (2.5)

The Common Q software test process is outlined in Section 7, “Software Test Plan,” of the SPM and is evaluated in Section 3.2.12 of this SE. This process includes module and unit testing performed during the implementation phases as well as integration, factory acceptance and site acceptance testing that are performed in the later phases of the Common Q software development life cycle. The integration and acceptance tests are performed with all application software installed into actual plant hardware so these tests are performed on the completed design implementation of the system.

System Features (2.5.1)

The testing performed on Common Q systems is intended to verify that all system requirements are validated. Because security requirements are integrated into the overall system requirements, they will also be validated by tests. Design validation is accomplished by the execution of integration, system, and acceptance tests. These tests are performed on the system configured as it is intended to be installed in the plant. Test configurations also include interfaces to other external systems.

Common Q system testing confirms that security controls are implemented and functioning to mitigate the corresponding vulnerabilities. In addition, vulnerability assessments are performed on the system during the test phase in order to identify the introduction of vulnerabilities or to confirm that no new vulnerabilities are introduced into the system. The NRC staff determined that the criterion of Regulatory Position C.2.5.1 of RG 1.152 has been met.

Development Activities (2.5.2)

Testing environments are isolated and maintained in accordance with the security program established by WEC. This program includes the establishment of an IDI to preclude inadvertent and remote access or changes that could affect the confidentiality or integrity of a system and related development system hardware or software. The NRC staff determined that the criterion of Regulatory Position C.2.5.2 of RG 1.152 has been met.

4.0 SUMMARY OF REGULATORY COMPLIANCE EVALUATIONS

On the basis of the foregoing review of the Common Q software development process for application software, the staff concludes that the SPM specifies plans that will provide a quality software life cycle process, and that these plans commit to documentation of life cycle activities that will permit the staff or others to evaluate the quality of the design features upon which the safety determination will be based. A review of the implementation of the life cycle process and the software life cycle process design outputs for specific applications will be performed on a plant-specific basis. This is addressed in Section 6.5 of the SE on LTR WCAP-16097-PINP Common Qualified Platform (ML12241A101).

On the basis of the review of WEC's software development process for application software, the staff concludes that the Common Q application development procedures will provide a quality software life cycle process, and that these plans commit to documentation of life cycle activities that will permit the staff or others to evaluate the quality of the design features upon which the safety determination will be based. The staff, therefore, concludes that the software program manual as applied to Common Q safety systems meets the guidance of RG 1.152 and that the special characteristics of computer systems have been adequately addressed. Based on its review, the staff finds, therefore, that the Common Q safety system software development processes when properly implemented are capable of producing software that will satisfy the requirements of GDC 1 and 21.

Cyber security to address malicious events is addressed under the purview of 10 CFR 73.54, "Protection of Digital Computer and Communication Systems and Networks," and thus has not been evaluated as part of this SPM review. Conformance to 10 CFR 73.54 is the responsibility of COL applicants or licensees who choose to reference the SPM.

4.1 Common Q SPM Generic Change Process

Per letter dated August 12, 2010 (Reference 6), WEC submitted WCAP-17266, "Common Q Platform Generic Change Process," (Reference 7) for NRC review and approval.

The Common Q generic change process defined by WCAP-17266 describes methods used by WEC to screen, and evaluate proposed changes to Common Q components, software or processes defined within the Common Q Platform and Software Program Manual topical reports subsequent to NRC review and approval. The scope of this process includes changes that are made to the Common Q SPM subsequent to the issuance of this SE. This process defines criteria to be used for the determination of whether the safety conclusions of the NRC safety evaluation remain valid following the proposed change or if the changes will require submittal to the NRC for evaluation and approval prior to implementation.

The staff has reviewed this document and acknowledges the benefits provided by implementation of a formal topical report screening, evaluation, and change process however, the NRC is unable to perform a safety evaluation of the processes defined by this document or make any safety conclusions regarding these processes at this time. This document is included as a reference within this safety evaluation in order to provide future reviewers of Common Q applications that reference this SE with information on how WEC evaluates and documents changes to the Common Q SPM. It is also beneficial for reviewers of Common Q applications to have access to the WEC generic change process in order to interpret the information provided in the Record of Changes document discussed below.

4.2 Common Q Record of Changes Document

Per letter dated August 25, 2010 (Reference 8), WEC submitted WCAP-16097, “Common Qualified Platform Record of Changes,” (Reference 9) for NRC review and approval.

The staff reviewed the Common Q Record of Changes (ROC) and confirmed that the changes to the Common Q SPM are consistent with the revised topical report evaluated by this SE. Furthermore, the staff reviewed the information provided in the Tables within the ROC and determined that these tables provide valuable information that should be used during application specific reviews to determine acceptability of changes to the Common Q SPM subsequent to the NRC review and approval of this License Topical Report (LTR). Plant-Specific action item 6 is therefore being included in this SE to provide direction for plant specific safety evaluations to include a review of the current Common Q record of changes to assess the validity of previously derived safety conclusions in light of the changes made to the Common Q SPM.

5.0 PLANT SPECIFIC ACTION ITEMS

An application may reference the approved WEC Common Q Topical Report provided the application satisfies the following conditions and limitations. The conditions and limitations are intended to ensure that all aspects of the digital safety system are properly designed and implemented. The following information is to be submitted or made available for staff audit/inspection upon receipt of an application for a license amendment, a design certification, or a combined license when referencing or incorporating by reference, TR WCAP-16096. The Common Q SPM and this safety evaluation provide the context and basis for the required additional information.

The following plant-specific actions must be performed by an applicant when requesting NRC approval for installation of a safety-related system based on the Common Q platform.

1. As noted in Sections 3.2.1 and 3.2.3, WEC may choose to use alternatives to the SPM defined processes when performing Initiation phase activities for individual projects. These alternatives are required to be documented in the Project Quality Plan (PQP). This PQP should be reviewed to determine if alternatives to the SPM are being used for development of project specific software. When such alternatives are being used, the PQP should be evaluated to determine if the justifications for the use of alternatives to the SPM processes are acceptable.

2. The Common Q SPM only includes the Software Life Cycle Process Planning Documentation as outlined in SRP BTP 7-14, Section B.2.1. As such, the plant-specific documentation outlined in SRP BTP 7-14, Sections B.2.2, “Software Life Cycle Process Implementation,” and B.2.3, “Software Life Cycle Process Design Outputs,” is to be evaluated separately for any application that references the Common Q SPM.

3. The Common Q SPM only addresses the vendor software planning processes for a Common Q-based system. For all activities in which the applicant or licensee assumes responsibility within a given project (including vendor oversight) for quality assurance, additional evaluations, audits or inspections must be performed to ensure that these licensee responsibilities are fulfilled.

4. Because the Common Q SPM does not address the criteria of BTP 7-14 Section B.3.1.8.4, “Software Operations Plan,” an evaluation of compliance must be

performed at the time of system development when the operational aspects of the system have been defined.

5. Site acceptance testing and installation testing are not covered under the Common Q Software Test Plan because they are considered to be licensee actions that are to be addressed during the development of a Common Q based application. As such, a project specific, site acceptance and installation test plan should be developed and used to address these aspects of software test planning. Because the Common Q SPM does not address all aspects of the BTP 7-14 Section B.3.2.4 criteria, an evaluation of compliance must be performed at the time of system development when the site and installation testing activities have been defined.

6. A licensee implementing an application based upon the Common Q platform should perform a review of the current Common Q Record of Changes document to assess the validity of previously derived safety conclusions if changes have been made to the Common Q SPM.

7. Secure Development and Operational Environment – An applicant or licensee referencing the Common Q SPM for a safety-related plant specific application should ensure that a secure development and operational environment has been established for its plant specific application, and that it satisfies the applicable regulatory evaluation criteria of RG 1.152, Revision 3.

6.0 REFERENCES

1. WCAP-16097-P-A, Revision 3, “Common Qualified Platform Topical Report” (Proprietary/Non-Proprietary), February 28, 2013, Agencywide Documents Access and Management System (ADAMS) Accession No. ML13112A110/ML13112A108.

2. Common Qualified Platform Topical Report WCAP-16097-P and 16097-NP Revision 3, June 30, 2012, ADAMS Accession Nos. ML12207A512 and ML12207A510.

3. Submittal of WCAP-16096-P and WCAP-16096-NP, Revision 4, “Software Program Manual for Common Qualified Platform,” (Proprietary/Non-Proprietary) for Review and Approval, July 17, 2012, ADAMS Accession No. ML12205A051.

4. Software Program Manual for Common Q Systems, Revision 4 WCAP-16096-P and 16096-NP Revision 4, June 30, 2012, ADAMS Accession Nos. ML12205A053 and ML12205A052.

5. WNA-DS-01070-GEN, Revision 6, “Application Restrictions for Generic Common Q Qualification,” December 31, 2011, ADAMS Accession Nos. ML11364A030 and ML11364A029.

6. Transmittal Letter for WCAP-17266-P/NP “Common Q Platform Generic Change Process,” Revision 0, August 12, 2010, ADAMS Accession No. ML102290175.

7. Common Q Platform Generic Change Process (WCAP-17266-P/NP), Revision 0, August 31, 2010, ADAMS Accession Nos. ML102290177 and ML102290176.

8. Transmittal Letter for WCAP-16097-P/NP “Common Qualified Platform Record of Changes”, Revision 1, April 19, 2012, ADAMS Accession No. ML12115A213.

9. Common Qualified Platform Record of Changes (WCAP-16097-P/NP), Revision 1, March 31, 2012, ADAMS Accession Nos. ML12115A215 and ML12115A214.

10. Software Program Manual for Common Q Systems (WCAP-16096-NP-A) Revision 1, January 29, 2004, ADAMS Accession No. ML040360115.

11. Safety Evaluation for Topical Report WCAP-16096-NP-A “Software Program Manual for Common Q Systems,” Revision 1, September 28, 2004, ADAMS Accession No. ML042730580.

12. Request for Additional Information, License Topical Report (WCAP-160096) “Software Program Manual for Common Q Systems,” September 29, 2011, ADAMS Accession No. ML112490485.

13. Response to NRC Request for Additional Information on WCAP-16096, Revision 2 “Software Program Manual for Common Q Systems,” January 31, 2012, ADAMS Accession No. ML12034A212.

14. Submittal of WCAP-16096-P/WCAP-16096-NP, Revision 5, “Software Program Manual for Common Q™ Systems” (Proprietary/Non-Proprietary), August 28, 2017, ADAMS Accession No. ML17241A112.

15. Request for Additional Information, License Topical Report (WCAP-160096) “Software Program Manual for Common Q Systems,” February 26, 2018, ADAMS Accession No. ML118018A005.

16. Response to NRC Request for Additional Information on WCAP-16096, Revision 5 “Software Program Manual for Common Q Systems,” May 31, 2018, ADAMS Accession No. ML18156A479.

17. Safety Evaluation for Topical Report WCAP-16096-P(NP)-A “Software Program Manual for Common Q Systems,” Revision 4, February 28, 2013, ADAMS Accession Nos. ML13081A046 and ML13081A047.

7.0 LIST OF ABBREVIATIONS

ABB	Asea Brown Boveri
AC160	Advant Controller 160
AF100	Advant Fieldbus 100
AISC	Application Specific Integrated Circuit
ALWR	Advanced Light Water Reactor
API	Application Programming Interface
ASME	American Society of Mechanical Engineers
ATWS	Anticipated Transients Without Scram
BIOB	Backplane I/O Bus
BTP	Branch Technical Position
CE	Combustion Engineering
CENP	Combustion Engineering Nuclear Power

CEA	Control Element Assembly
CEAC	Control Element Assembly Calculator
CEAPD	CEA Position Display
CENP	CE Nuclear Power (Westinghouse)
CEO	Cognizant Engineering Organization
CETMS	Core Exit Thermocouple Monitoring System
CGD	Commercial-Grade Dedication
Common Q	Common Qualified
COTS	Commercial-Off-The-Shelf
CPC	Core Protection Calculator
CPCS	Core Protection Calculator System
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CS	Communication Section
CWP	CEA Withdrawal Prohibit
D-in-D&D	Defense in Depth and Diversity
DB	Database
DBE	Design Basis Event
DESFAS	Digital ESFAS
DI	Digital Input
DLCE	Design Life Cycle Evaluation
DNBR	Departure from Nucleate Boiling Ratio
DPPS	Digital Plant Protection System
DPRAM	Dual Port Random Access Memory
DSP	Data Set Peripheral
EIA	Electronic Industries Association
EMC	Electromagnetic Compatibility
EPRI	Electric Power Research Institute
EPLD	Erasable Programmable Logic Device
ESF	Engineered Safety Features
ESFAS	Engineered Safeguards Features Actuation System
FAT	Factory Acceptance Test
FCB	Function Chart Builder
FE	Function Enable
FMEA	Failure Modes and Effect Analysis
FOM	Fiber Optic Modem
FPD	Flat Panel Display
FPDS	Flat-Panel Display System
FSAR	Final Safety Analysis Report
GDC	General Design Criteria
GUI	Graphical User Interface
HDD	Hard Disk Drive
HDLC	High Level Data Link Control
HJTC	Heated Junction Thermocouple
HMI	Human Machine Interface
HSI	Human System Interface
HSL	High Speed Link
I/O	Input/Output
I&C	Instrumentation and Control
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers

IPC	Interprocess Communication
ISR	Interrupt Service Routine
ITP	Interface and Test Processor
IVV	Independent Verification And Validation
LC	Loop Controller
LCLP	Local Coincidence Logic Processor
LED	Light Emitting Diode
LPD	Local Power Density
MCR	Main Control Room
MTBF	Mean Time Between Failures
MTP	Maintenance and Test Panel
NEI	Nuclear Energy Institute
NSSS	Nuclear Steam Supply System
OM	Operator's Module
OBE	Operational Basic Earthquake
PAMS	Post-accident Monitoring System
PAS	Plant Annunciator System
PC	Process Control
PCB	Printed Circuit Board
PCE	Program Control Element
PDS	Previously Developed Software
PIT	Precision Interval Timer
PLC	Programmable Logic Controller
PM	Processor Module
PPS	Plant Protection System
PROM	Programmable Read-only Memory
PS	Processing Section
QA	Quality Assurance
QSPDS	Qualified Safety Parameter Display System
RAM	Random Access Memory
RCM	Remote Control Module
RCP	Reactor Coolant Pump
RFI	Radio Frequency Interference
RG	Regulatory Guide
RPS	Reactor Protection System
RSP	Remote Shutdown Panel
RSPT	Reed Switch Position Transmitter
RTC	Real Time Clock
RTD	Resistance Temperature Detector
RTS	Reactor Trip System
RTCB	Reactor Trip Circuit Breaker
RVLMS	Reactor Vessel Level Monitoring System
SAR	Safety Analysis Report
SBC	Single Board Computer
SCADA	Supervisory Control and Data Acquisition
SCMP	Software Configuration Management Plan
SCR	Software Change Request
SDM	Service Data Manager
SDP	Service Data Protocol
SE	Safety Evaluation
SLC	Software Life-Cycle

SLE	Software Load Enable
SMM	Subcooled Margin Monitor
SPM	Software Program Manual
SQAP	Software Quality Assurance Plan
SRAM	Static RAM
SRP	Standard Review Plan
SSP	Software Safety Plan
STS	Standard Technical Specifications
SVVP	Software Verification and Validation Plan
SW	Software
SWC	Surge Withstand Capability
TCB	Task Control Block
TMI	Three Mile Island
TS	Technical Specification(s)
TSTF	Technical Specification Task Force
V&V	Verification and Validation
WWDT	Window Watchdog Timer

Appendix A - Comments on Draft Safety Evaluation and NRC Staff Resolution

Comment Number	Comment Location	Comment Type	Comment	NRC Response
1	Page 1/Line 22	Editorial	“Reference 0” should be “Reference 14”	Agree. Reference 14 is Rev. 5 submittal letter for the SPM.
2	Page 3/Line 25	Editorial	Should “Revision 1” be added to RG 1.170?	Yes, Add “Revision 1” to reference.
3	Page 6/Lines 10 – 11	Editorial	Westinghouse (WEC) suggests changing “the ABB Master Programming Language Control Configuration (ACC) and Photon” to “approved.” WEC would like to remove references to specific software tools (e.g., AMPL and Photon) because the SPM does not specifically cite these tools; they are only cited in the Common Qualified Platform Topical Report (WCAP-16097).	Agree to delete the specific tool name references however, we do not want to imply that the tools are NRC approved. Change to the following: “Software will be developed using WEC approved software development tools.”

Comment Number	Comment Location	Comment Type	Comment	NRC Response
4	Page 12/Lines 24 – 26	Clarification	The SER states that the "SQAP no longer applies to software classified as: important to availability or general purpose software." However, this is not consistent with the text in the SQAP, Section 4.1.2, "Scope," which states: "This SQAP is required for all quality classifications defined for the Common Q™ system: protection, important-to-safety, important-to-availability, and general purpose software." Therefore, WEC suggests reverting to the wording from the previous revision of the SER: "The scope of the Common Q SQAP includes software in all four SIL classifications; protection, important to safety, important to availability, and general purpose. The Common Q SQAP applies to original software that was developed under the requirements of the Common Q SPM."	Agree. This was a carryover from the original WCAP-16096, Revision 5 submittal which had removed the other SIL classifications from scope. The WEC response to RAI 1.c. reinstated these SIL levels to the SPM scope. Change as edited in this document.
5	Page 15/Line 29	Editorial	"Reference 0" should be "Reference 16"	Agree. Reference 16 is the WEC response to RAIs so Reference 16 is correct.

Comment Number	Comment Location	Comment Type	Comment	NRC Response
6	Page 16/Line 7	Editorial	WEC suggest changing “the AMPL Control Configuration (ACC)” to “an approved”. WEC would like to remove references to specific software tools because the SPM does not specifically cite these tools; they are only cited in the Common Qualified Platform Topical Report (WCAP-16097).	Agree to delete the specific tool name references however, we do not want to imply that the tools are NRC approved. Change to the following: “For a Common Q project this level of integration is accomplished by the creation of control functions using a WEC approved, development tool.”
7	Page 16/Line 8	Editorial	WEC suggests changing “ACC” to “the tool”. WEC would like to remove references to specific software tools because the SPM does not specifically cite these tools; they are only cited in the Common Qualified Platform Topical Report (WCAP-16097).	Agree. Change to the following: “Proper use of this tool involves assembly of pre-approved Program Control (PC) elements into complete control functions.”
8	Page 16/Lines 16 - 17	Editorial	WEC suggests changing “the photon” to “an approved”. WEC would like to remove references to specific software tools because the SPM does not specifically cite these tools; they are only cited in the Common Qualified Platform Topical Report (WCAP-16097).	Agree to delete the specific tool name references however, we do not want to imply that the tools are NRC approved. Change to the following: “FPDS software applications are developed using a WEC approved graphical user interface software tool.”

Comment Number	Comment Location	Comment Type	Comment	NRC Response
9	Page 16/Lines 27 – 28	Editorial	WEC suggests deleting “using the ACC tool”. WEC would like to remove references to specific software tools because the SPM does not specifically cite these tools; they are only cited in the Common Qualified Platform Topical Report (WCAP-16097).	Agree to delete the specific tool name references however, we do not want to imply that the tools are NRC approved. Change to the following: “The system hardware architecture is established in conjunction with the application software using a WEC approved tool;
10	Page 18/Lines 38 – 39	Clarification	WEC suggests adding “if within Westinghouse’s scope of supply” to the end of the first sentence because creating training materials for end users may not be in Westinghouse’s supply contract. This is clarified in section 5.5.7.2 of the SPM, which states: “Review training materials (if within Westinghouse’s scope of supply) for the following:”	Agree. Change as edited in this document.
11	Page 22/Line 30	Editorial	WEC suggests changing “high, major, moderate, and low” to “4, 3, 2, and 1”. IEEE Std. 1012-2004 now uses “4, 3, 2, and 1” for their software integrity level scheme.	Agree. Change as edited in this document.
12	Page 30/Line 2	Editorial	WEC suggests changing “System Security Capabilities” to “Secure Operational Environment Capabilities” to be consistent with the revised heading in the SDOE section of the SPM.	Agree. Change to “Secure Operational Environment Capabilities” as suggested.

Comment Number	Comment Location	Comment Type	Comment	NRC Response
13	Page 30/Line 29	Editorial	WEC suggests changing “Identification of Life Cycle Vulnerabilities” to “General Life Cycle Vulnerabilities” to be consistent with the revised heading in the SDOE section of the SPM.	Agree. Change to “General Life Cycle Vulnerabilities” as suggested.
14	Page 30/Lines 38 - 40	Clarification	WEC suggests changing “The SPM calls for a software life cycle vulnerabilities assessment V&V activities to be performed during the Concept, Requirements, Design and Test phases.” to “The SPM calls for V&V activities to be performed during the Concept, Requirements, Design, Implementation, and Test phases to verify correct implementation of secure operational environment requirements.” This revision better aligns with the revised SDOE section.	Agree. Change as edited in this document.
15	Page 30/Lines 40 – 41	Clarification	WEC suggests deleting “The SPM also identifies human factors to be used for mitigation of system vulnerabilities.” This revision better aligns with the revised SDOE section.	Agree. Delete following sentence: “The SPM also identifies human factors to be used for mitigation of system vulnerabilities.”

Comment Number	Comment Location	Comment Type	Comment	NRC Response
16	Page 30/Lines 44 – 46	Clarification	WEC suggests changing “Subsequent assessments are also performed during the requirements, design, implementation and test phases.” to “These vulnerabilities become platform restrictions that are confirmed through the design, implementation, and test phases.” This revision better aligns with the revised SDOE section.	Staff does not agree that all identified vulnerabilities need to become platform restrictions. The point of this assessment is to ensure that processes will identify and address vulnerabilities that might be introduced to the system during later stages of the development process. Change to the following: “Subsequent assessments are also performed to determine if new vulnerabilities are introduced to the system during the later stages of the development process.”
17	Page 31/Line 40	Clarification	WEC suggests changing “requirements phase” to “concept phase” in order to better align with the revised SDOE section.	Agree to change “requirements” to “concept.”
18	Page 32/Line 41	Editorial	“IEEE Std. 1028-2005” should be “IEEE Std. 1028-2008”	Agree. Confirmed this reference should be IEEE 1028-2008.

Comment Number	Comment Location	Comment Type	Comment	NRC Response
19	Page 34/Lines 20 – 22	Clarification	WEC suggests changing "The V&V activities to be performed during the implementation phase include performing a security assessment of the system to verify that the security controls chosen in the design phase are adequate" to "The V&V activities to be performed during the implementation phase verify that the security controls chosen in the design phase have been properly implemented". This revision better aligns with the revised SDOE section.	Just saying that security controls from design phase are properly implemented ignores the possibility that new vulnerabilities could be introduced and/or identified during design implementation. For this reason, we expect a vulnerability analysis V&V task to be performed during implementation. IEEE 1012-2004 includes performance of a Security Analysis during each stage of development as a minimum required V&V task. The security analysis is also included for each phase as indicated in Table 2 (Exhibit 5-8).
20	Page 34/Lines 22 – 24	Clarification	WEC suggests deleting "If system vulnerabilities are identified during this security assessment then requirements for additional security controls are added to the system requirements in order to address or otherwise mitigate these vulnerabilities" in order to better align with the revised SDOE section.	Staff does not agree with this deletion. The NRC expects WEC to take appropriate actions to address any new vulnerabilities that might be introduced during design implementation. By deleting this sentence, aren't we saying that WEC doesn't have to address these newly identified vulnerabilities?

Comment Number	Comment Location	Comment Type	Comment	NRC Response
21	Page 35/Lines 30 – 32	Clarification	WEC suggests deleting “In addition, Vulnerability assessments are performed on the system during the test phase in order to identify the introduction of vulnerabilities or to confirm that no new vulnerabilities are introduced into the system” in order to better align with the revised SDOE section.	Exhibit 5-8, Table 2, “Minimum V&V tasks assigned to each software integrity level” includes performance of a security analysis V&V activity at each stage of development including test. This is required for SIL 4 software and therefore is required for Common Q Protection software.
22	Page 38/Lines 11 – 15	Clarification	Normally a PSAI is cited in the text of the SER, but it's not in this case. Having corresponding text in the SER helps provide context for the reason behind the PSAI.	Add the following sentence to the end of the second paragraph in Section 3.2.13: “In addition, an applicant or licensee using a Common Q platform based system must perform actions to satisfy PSAI 7.”
23	Page 39/Line 33	Editorial	WEC suggests deleting the “ACC” acronym since it is not cited in the SER text.	Agree. Delete the ACC acronym.
24	Page 39/Line 37	Editorial	WEC suggests deleting the “AMPL” acronym since it is not cited in the SER text.	Agree. Delete AMPL acronym.
25	Page 40/Line 39	Editorial	“HIS” should be changed to “HSI”	Agree. Microsoft Word automatically changes this to HIS.
26	Page 41/Line 22	Editorial	WEC suggests deleting the “QSSL” acronym since it is not cited in the SER text.	Agree. Delete acronym.

Comment Number	Comment Location	Comment Type	Comment	NRC Response
27	Page 42/Lines 13 – 14	Editorial	WEC suggests deleting “QNX® and Photon® are registered trademarks of QNX Software Systems GmBH & Co. KG ("QSSKG", formerly "QSSL") and are used under license by QSS” since “QNX” and “Photon” are not used in the SER.	Agree to delete text as suggested.

Section B

LTR-NRC-18-36, “Responses to NRC Request for Additional Information for Westinghouse Topical Report WCAP-16096-P/WCAP-16096-NP, Revision 5, "Software Program Manual for Common Q™ Systems" (ADAMS Accession No. ML18156A479)

Westinghouse Electric Company
1000 Westinghouse Drive
Cranberry Township, Pennsylvania 16066
USA

U.S. Nuclear Regulatory Commission
Document Control Desk
11555 Rockville Pike
Rockville, MD 20852

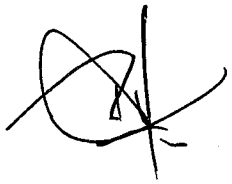
Direct tel: (412) 374-5130
Direct fax: (724) 940-8542
e-mail: hosackkl@westinghouse.com

LTR-NRC-18-36

May 31, 2018

Subject: Responses to NRC Request for Additional Information for Westinghouse Topical Report WCAP-16096-P/WCAP-16096-NP, Revision 5, "Software Program Manual for Common Q™ Systems" (Docket No.: 99902038; EPID: L-2017-TOP-0059)

Enclosed is a copy of the Responses to NRC Request for Additional Information for Westinghouse Topical Report WCAP-16096-P/WCAP-16096 NP, Revision 5, "Software Program Manual for Common Q™ Systems". This submittal contains responses for the thirteen RAIs transmitted via NRC letter to James A. Gresham (Westinghouse) dated February 26, 2018 (ML18018A005). All content in the Responses is non-proprietary, and as such, only a non-proprietary version is provided.



Korey L. Hosack, Manager
I&C Licensing & Regulatory Support

Enclosures

cc: Joseph Holonich
Dennis Morey

Responses to NRC Responses to NRC Request for Additional Information for Westinghouse Topical Report WCAP-16096-P/WCAP-16096 NP, Revision 5, "Software Program Manual for Common Q™ Systems"

(Non-Proprietary)

May 2018

Westinghouse Electric Company
1000 Westinghouse Drive
Cranberry Township, PA 16066

REQUEST FOR ADDITIONAL INFORMATION

WCAP-16096-P, "SOFTWARE PROGRAM MANUAL FOR COMMON Q SYSTEMS"

1. Compliance with Institute of Electrical and Electronics Engineers Standard 1012

Title 10, "Energy" of the *Code of Federal Regulations* (CFR) Part 50 requires in Appendix B, "Quality Assurance Criteria for Nuclear Power Plants and Fuel Reprocessing Plants," in part in Criterion II, "Quality Assurance Program," that, "The [quality assurance] program shall take into account the need for special controls, processes, test equipment, tools, and skills to attain the required quality, and the need for verification of quality by inspection and test." Additionally, in Criterion III, "Design Control," it requires, in part, that, "These measures shall include provisions to assure that appropriate quality standards are specified and included in design documents and that deviations from such standards are controlled…." The design control measures shall provide for verifying or checking the adequacy of design, such as by the performance of design reviews, by the use of alternate or simplified calculation methods, or by the performance of a suitable testing program.

The staff endorsed a method found to be acceptable when performing the verification and validation (V&V) activities associated with the development of a safety-related software based system via Revision 2 of Regulatory Guide (RG) 1.168, "Verification, Validation, Reviews, and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants." In the RG, it endorses the Institute of Electrical and Electronic Engineers (IEEE) Standard (Std.) 1012-2004, "IEEE Standard for Software Verification and Validation."

Previous versions of WCAP-16096-P, "Software Program Manual for Common Q Systems" (SPM), up to and including Revision 4, stated that its Software Verification and Validation (SVV) Plan (SVVP) complied with the IEEE Std. 1012, "IEEE Standard for Software Verification and Validation," whether the 1986 or 1998 version – dependent upon the revision of the SPM. This compliance statement was used as a partial basis for the acceptability of the SPM in the original and subsequent safety evaluations (SEs) related to the method of software system development described in the SPM. In Revision 5 of the SPM, the compliance statement to IEEE Std. 1012-2004 has been removed.

The changes made in Revision 5 of the SPM appear to indicate that the SVVP will no longer be required to comply with IEEE Std. 1012-2004. As highlighted in the examples below, please clarify and provide additional information on the revised approach to developing application level software for the Common Q System without compliance to IEEE Std. 1012-2004, along with the basis and justification.

If the SPM intends to take exception to the requirements of IEEE Std. 1012 for V&V activities, then please provide sufficient justification (inputs, tasks/activities, and outputs) at a similar level of decomposition and granularity within IEEE Std. 1012 to demonstrate an alternative approach that complies with 10 CFR Part 50, Appendix B. In addition, clarify if the SPM is taking exception to compliance with IEEE Std. 7-4.3.2 and, if so, provide similar justification.

Westinghouse Response:
Westinghouse does not intend to take exception to the requirements for V&V activities in IEEE Std. 1012-2004. The SPM complies with IEEE Std. 1012-2004 requirements for V&V activities as documented in Exhibit 5-8 of the SPM. Therefore, Westinghouse will update the SPM throughout to indicate that it complies with the requirements for V&V activities in IEEE Std. 1012-2004. Accordingly, Westinghouse does not intend to take exception to IEEE Std. 7-4.3.2-2003.

a. Section 3.3.9, "Software Verification and Validation Activities" – Reference 8, [IEEE Std. 1012 – 2004], is no longer included in the compliance statement. Please clarify if the V&V activities in this area are taking exception to this standard or taking a different approach for meeting the requirements of 10 CFR Part 50, Appendix B.

Westinghouse Response:
The SPM complies with IEEE Std. 1012-2004 requirements for V&V activities as documented in Exhibit 5-8 of the SPM. Therefore, Section 3.3.9, "Software Verification and Validation Activities," will be revised to say (as originally stated in Revision 4):
"These activities conform to the requirements in References 8 and 11."

b. Section 5.1, "Purpose," of the SVVP – the IEEE Std. 1012 compliance statement has been removed. Beginning in Revision 3 of the SPM, along with the information contained in Exhibit 5.8, "IEEE Standard 1012-1998 Compliance Table" [IEEE Std. 1012-2004 version for Revision 5 of the SPM], that explained where in the SPM the related sections of IEEE Std. 1012 could be located, the staff relied on the more detailed information within IEEE Std. 1012 describing exactly what and how Independent Verification and Validation (IV&V) inputs, tasks/activities, and outputs would be conducted. Please provide a list of what SVV activities and tasks will no longer be conducted as described in Table 1 – "V&V Tasks, Inputs and Outputs" of IEEE Std. 1012 and justification for why the given tasks, inputs, and outputs are no longer required.

Westinghouse Response:
The SPM complies with IEEE Std. 1012-2004 requirements for V&V activities as documented in Exhibit 5-8 of the SPM. Therefore, Section 5.1, "Purpose," will be revised to say:

"This section explains requirements for the IV&V processes starting with the system design document stage and all necessary IV&V activities to verify and/or validate I&C systems. This SVVP complies with Reference 8 requirements for V&V activities."

c. Table 5.9-1 identifies both 'Important to Availability' and 'General Purpose' software as being, 'IEEE Std. 1012 Not Applicable.' The NRC staff previously determined these classifications to be compliant with IEEE Std. 1012 because V&V tasks for these classifications were defined in Exhibit 5-8. Since there has been no corresponding change to remove 'Important to Availability' or 'General Purpose' software classifications from Exhibit 5-8, please provide a list of what V&V activities that are no longer considered to be compliant with IEEE 1012 and the reasoning behind such changes.

Westinghouse Response:
As allowed by IEEE Std. 1012-2004, software classified as SIL 1 and SIL 2 can follow a subset of the V&V activities required for SIL 4 software. Common Q Software classified as General Purpose maps to SIL 1, while software classified as Important to Availability maps

to SIL 2. Exhibit 5-1 provides a listing of the V&V activities that will be performed for ITA and General Purpose software. Therefore, Table 5.9-1 will be updated as follows:

Table 5.9-1. Software Classification Mapping

SPM Classification	IEEE Standard 1012-2004
Protection	4
Important-to-Safety	4 (with noted exceptions identified in **EXHIBIT 5-8 IEEE STANDARD 1012-2004 COMPLIANCE TABLE)**
Important-to-Availability	~~N/A – V&V of non-safety systems is not in accordance with IEEE Std. 10122~~ – See EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES
General Purpose	~~N/A – V&V of non-safety systems is not in accordance with IEEE Std. 10121~~ – See EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES

In order to address the NRC's concern over requirements traceability, Section 5.5.3.2, "IV&V Tasks," will be revised as follows:

"The following are specific IV&V Tasks:

1. Review the adequacy and accuracy of the Requirements Traceability Matrix (RTM) as prepared by the design team. The traceability in the RTM is established in both directions at each decomposition level and allows IV&V to verify the software requirements are complete, correct, and accurate decomposition of allocated system requirements. The review shall include verification that all functional, hardware interface, software, performance, and user requirements have been included."

d. Section 10.5, "Software Verification and Validation Documentation" - The IEEE Std. 1012 compliance statement has been removed from this section and replaced by a reference to Section 5.6, "Software Verification and Validation Reporting," of the SPM. Section 5.6 does not contain an IEEE Std. 1012 compliance statement. Please clarify what V&V activities in this area are taking exception to the standard or taking a different approach for meeting the requirements of 10 CFR Part 50, Appendix B.

Westinghouse Response:

Westinghouse does not intend to take exception to the reporting requirements of IEEE Std. 1012-2004. Therefore, Section 10.5 will be revised as follows (as originally stated in Revision 4):

"Software IV&V documentation shall include Software IV&V Reports (SVVR), prepared according to ~~Section 5.6~~Reference 8 as augmented by Reference 18."

e. IEEE 7-4.3.2 2003, "IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations," states, in part, "…the software V&V effort shall be performed in accordance with IEEE Std. 1012." Since, in Section 3.3.9, "Software Verification and Validation Activities," the SPM states that "These activities conform to the requirements in Reference 11," which is IEEE Std. 7-4.3.2. IEEE Std. 7-4.3.2 also requires compliance with IEEE Std. 1012. Please clarify if V&V activities in this area are taking exception to this standard or taking a different approach for meeting the requirements of 10 CFR Part 50, Appendix B.

Westinghouse Response:
The SPM complies with IEEE Std. 1012-2004 requirements for V&V activities as documented in Exhibit 5-8 of the SPM. Therefore, the SPM complies with IEEE Std. 7-4.3.2-2003.

2. Compliance with IEEE Standard 829 Requirements

The regulation at 10 CFR Part 50, Appendix B requires, in part, that, "The [quality assurance] program shall take into account the need for special controls, processes, test equipment, tools, and skills to attain the required quality, and the need for verification of quality by inspection and test." Additionally, in Criterion III, "Design Control," it requires, in part, that, "These measures shall include provisions to assure that appropriate quality standards are specified and included in design documents and that deviations from such standards are controlled…." The design control measures shall provide for verifying or checking the adequacy of design, such as by the performance of design reviews, by the use of alternate or simplified calculation methods, or by the performance of a suitable testing program.

The staff endorsed a method found to be acceptable when performing the testing and documenting the test activities associated with the development of a safety-related software based system via Revision 1 of RG 1.170, "Test Documentation for Digital Computer Software used in Safety Systems of Nuclear Power Plants." In the RG, it endorses the IEEE Std. 829-2008, "IEEE Standard for Software and System Test Documentation."

Previous versions of the SPM, including Revision 4, stated that the SVVP complied with IEEE Std. 829. As highlighted in the examples below, which describe how the test plans, procedures, test summary reports, and other SVV test documentation will be managed, it appears to indicate that testing documentation will no longer be required to comply with IEEE Std. 829-2008 [or in some cases in content, but not necessarily in format]. For each example below, please clarify if documentation activities in this area are taking exception to this standard or taking a different approach for meeting the requirements of 10 CFR Part 50, Appendix B.

If the SPM intends to take exception to some or all of the requirements of IEEE Std. 829, then please provide sufficient justification (inputs, tasks/activities, and outputs) at a similar level of decomposition and granularity within IEEE Std. 829 to

demonstrate how an alternative approach complies with the requirements of 10 CFR Part 50, Appendix B.

Westinghouse Response:
Westinghouse intends to take exception to IEEE Std. 829-2008 as endorsed by Regulatory Guide (RG) 1.170, Rev. 1, and instead will use an alternative approach that complies with the requirements of 10 CFR Part 50, Appendix B. To do so, Westinghouse will revise the SPM to state compliance to the previously cited RG revision (i.e., Rev. 0) and IEEE Std. 829-1998. This older RG meets the same underlying regulatory criteria (i.e., GDC 1 and 21 of Appendix A to 10 CFR Part 50, as well as Criteria I, II, III, V, VI, XI, and XVII of Appendix B) as the new RG. As a result, this alternative approach will meet the underlying regulatory criteria of RG 1.170, Rev. 1. Therefore, Reference 14 will be revised as follows (as originally stated in Revision 4):

"IEEE Std 829-~~2008~~1998, "IEEE Standard for Software ~~and System~~ Test Documentation""

And Reference 20 will be revised as follows (as originally stated in Revision 4):

"Reg. Guide 1.170, Rev. ~~10~~ (~~July 2013~~Sept. 1997), "Software Test Documentation for Digital Computer Software used in Safety Systems of Nuclear Power Plants""

a. Section 4.3.2.2, "Software Requirements Phase," of Revision 5 of the SPM states, in part, "A Common Q [Qualification] specific test plan shall start to be developed to identify how the test activities will be implemented. Reference 14 [IEEE Std. 829-2008], Section 8 will be used as guidance in developing the test plan." However, in Revision 4 of the SPM the Common Q specific test plan shall start to be developed in accordance with the content, but not the format of Reference 14 [IEEE Std. 829-1998], Section 7, "Test Procedure Specification," and Section 11, "Test Summary Report," respectively.

   Westinghouse Response:
   As stated above, the SPM will be revised to state compliance to the previously cited RG revision (i.e., Rev. 0) and IEEE Std. 829-1998 as an alternate approach to meet the underlying regulations of RG 1.170, Rev. 1. Therefore, Section 4.3.2.2, "Software Requirements Phase," will be revised as follows (as originally stated in Revision 4):

   "A Common Q™ specific test plan shall start to be developed in accordance with the content, not the format of Reference 14, Section 4, to identify how the test activities will be implemented. ~~Reference 14, Section 8 will be used as guidance in developing the test plan. The test plan shall comply with the requirements of Reference 1 and Reference 4.~~ It shall include the following topics as a minimum:"

b. Section 4.5.2.2, "Software Testing Standards," of Revision 5 of the SPM states, in part, "Specific format and content for test procedures and test reports shall also be provided in the Test Plan and shall comply with Section 5.8 [of the SPM]." In Revision 4 of the SPM, it states, in part, "Specific format and content for test procedures and test reports shall also be provided in the Test Plan and shall comply with Reference 14 [IEEE Std. 829-1998] Sections 7 and 11 ['Test Procedure Specification' and 'Test Summary Report' respectively]." However, in Revision 5 of the SPM, Section 5.8 does not contain an IEEE Std. 829 compliance statement.

Westinghouse Response:
As stated above, the SPM will be revised to state compliance to the previously cited RG revision (i.e., Rev. 0) and IEEE Std. 829-1998 as an alternate approach to meet the underlying regulations of RG 1.170, Rev. 1. Therefore, Section 5.8, "IV&V Test Documentation Requirements," will be revised as follows (as originally stated in Revision 4):

"The purpose of this section is to define the purpose, format and content of required test documentation. The test documentation as a whole shall fulfill the requirements of References 14 and 20 ~~are used as guidance in creating the test documentation~~. ~~The Test Documentation shall be in accordance with the NRC-accepted Westinghouse 10 CFR 50, Appendix B Quality Management System (Reference 1) and quality assurance procedures (Reference 4).~~"

Section 5.8.2, "Test Procedure," will be revised as follows (as originally stated in Revision 4):

"The elements of the test specification and test cases described in Reference 14 can be found in the test procedure. ~~Reference 14, Section 12 will be used as guidance in developing the test procedures.~~ The test procedure shall comply with ~~the requirements of Reference 1 and~~ Reference 14, Section 7."

Section 5.8.3, "Test Report," will be revised as follows (as originally stated in Revision 4):

"The test report also contains the Exception Report log and copies of the Exception Reports. Together, these identify the status of outstanding test exceptions reported during testing. ~~Reference 14, Section 16 will be used as guidance in developing the test reports.~~ The test reports shall comply with ~~the requirements of Reference 1 and~~ Reference 14, Section 11.

c. Section 5.4.5.2 "IV&V Core Activities," Item 3 and Item 4 replace compliance commitment to documentation requirements of IEEE Std. 829-2008, with a reference to Section 5.8 of the SPM. However, in Revision 5 of the SPM, Section 5.8 does not contain an IEEE Std. 829 compliance statement.

Westinghouse Response:
As stated in Westinghouse's response to RAI 2.b, Section 5.8 will be revised to state compliance to the previously cited RG revision and IEEE Std. 829-1998 as an alternate approach to meet the underlying regulations of RG 1.170, Rev. 1.

d. Section 5.5.3.2, "[Requirements Phase] IV&V Tasks," V&V Task 10 (Task 9 in Revision 4 of the SPM) replaces the compliance commitment to test plan development requirements of IEEE Std. 829 with a reference to Section 4.3.2.2 of the SPM. In Revision 5 of the SPM, Section 4.3.2.2 no longer contains an IEEE Std. 829 compliance statement. Instead it replaced the previous compliance statement in Revision 4 of the SPM with a statement that IEEE Std. 829 will be used as guidance in developing the test plan.

Westinghouse Response:
As stated in Westinghouse's response to RAI 2.a, Section 4.3.2.2 will be revised to state compliance to the previously cited RG revision and IEEE Std. 829-1998 as an alternate approach to meet the underlying regulations of RG 1.170, Rev. 1.

e. Section 5.5.4.2, "[Design Phase] IV&V Tasks," Item 9 replaces the compliance commitment to test procedure development requirements of IEEE Std. 829 with a reference to Section 5.8 of the SPM. However, in Revision 5 of the SPM, Section 5.8 does not contain an IEEE Std. 829 compliance statement.

Westinghouse Response:
As stated in Westinghouse's response to RAI 2.b, Section 5.8 will be revised to state compliance to the previously cited RG revision and IEEE Std. 829-1998 as an alternate approach to meet the underlying regulations of RG 1.170, Rev. 1.

f. Section 9.3.2.2, "Detailed Analysis," replaces the compliance content commitment to test plan requirements of IEEE Std. 829 with a reference to Section 4.3.2.2 of the SPM. When compared to Revision 4 of the SPM, Section 4.3.2.2 no longer contains an IEEE Std. 829 compliance statement. Instead it replaced the previous compliance statement with a statement that IEEE Std. 829 will be used as guidance in developing the test plan.

Westinghouse Response:
As stated in Westinghouse's response to RAI 2.a, Section 4.3.2.2 will be revised to state compliance to the previously cited RG revision and IEEE Std. 829-1998 as an alternate approach to meet the underlying regulations of RG 1.170, Rev. 1.

g. Section 9.4.2, "Design Process," replaces the compliance commitment to test procedure development requirements of IEEE Std. 829 with a reference to Section 5.8 of the SPM. However, when compared to Revision 4 of the SPM, Section 5.8 does not contain an IEEE Std. 829 compliance statement.

Westinghouse Response:
As stated in Westinghouse's response to RAI 2.b, Section 5.8 will be revised to state compliance to the previously cited RG revision and IEEE Std. 829-1998 as an alternate approach to meet the underlying regulations of RG 1.170, Rev. 1.

h. Section 9.6.2, "Test Process," replaces the compliance commitment to test procedure development requirements of IEEE Std. 829 with a reference to Section 5.8 of the SPM. However, when compared to Revision 4 of the SPM, Section 5.8 does not contain an IEEE Std. 829 compliance statement.

Westinghouse Response:
As stated in Westinghouse's response to RAI 2.b, Section 5.8 will be revised to state compliance to the previously cited RG revision and IEEE Std. 829-1998 as an alternate approach to meet the underlying regulations of RG 1.170, Rev. 1.

3. **Preparation of Site Test Plan**

In Revision 5 to the SPM, Section 4.3.2.6 includes a change to the process for development of a site test plan which allows development of such a plan to occur at a later

stage of the development lifecycle to support evaluation of requirement testability on-site. There does not appear to be a corresponding change to the V&V activities associated with this issue.

In Revision 4 to the SPM, the preparation of a site test plan occurred during the requirements phase, which is consistent with the requirements of IEEE Std. 1012. This was reflected in Exhibit 5-8 as an item titled "Acceptance V&V Test Plan Generation" and this test plan covered acceptance, integration, system, and component levels. The staff needs to understand where this site testing activity now fits in relation to the V&V activities in the "IEEE Standard 1012 – 2004 Compliance Table" (Exhibit 5-8) now that an allowance for Site Acceptance Test Plan development at a later stage is described. Please provide additional information to identify the specific V&V activity requirement for development of the Site Acceptance Test Plan. Provide a discussion of when the required activity is to be performed in relation to the development lifecycle, and why doing so at that particular phase of system development is acceptable.

Westinghouse Response:
IEEE Std. 1012-2004 does not differentiate between a Factory Acceptance Test (FAT) plan and a Site Acceptance Test (SAT) plan. Therefore, the FAT plan will be generated during the requirements phase as shown in Exhibit 5-8 of the SPM. If Westinghouse is contracted to perform site acceptance testing, Westinghouse will work with the Licensee to develop the required inputs for the SAT plan. The contract schedule will then define when the SAT plan will be developed. Therefore, the text for "Acceptance V&V test plan generation" in Exhibit 5-8 will be revised as follows:

"One test plan covers ~~Acceptance, integration, system, and componentall~~ phases of testing, except SAT. A separate SAT plan will be developed in accordance with the contract schedule with the licensee."

4. **Testing Sequence** - Section 7.2.4 of the SPM now includes provisions for deferring completion of test activities to allow commencement of the subsequent tests before the preceding test level is complete. Please provide additional information to explain why these new provisions for the testing sequence are being made and provide justification for allowing testing levels to proceed in a sequence other than previously prescribed.

Westinghouse Response:
A software module can be generically produced (existing software not to be modified) or maybe specifically developed or modified for a particular project (new software, or existing software to be modified). In the former, pre-validated modules are used in the application software and the project's validation testing starts with Unit Testing of the released application. In the latter case, however, the validation of the software module (module test) can be performed while the application software that uses the module is concurrently undergoing downstream validation tests. This is a calculated risk in the project execution where rework in the downstream validation activities may be required should the module test failed. Nevertheless, the scope of module test, or any downstream test activities, is not changed due to this provision.

5. **Deferral of Factory Acceptance Test Activities to Site**

Section 7.3.1.5, “Factory Acceptance Test (FAT),” of the revised SPM now allows for deferral of FAT activities to be conducted at the site following installation. Considering the stated objective of the FAT as demonstrating that the complete system is integrated and functional, it is unclear how these objectives will be achieved prior to shipment of equipment to the site when FAT activities are deferred. Please provide additional information describing how FAT objectives will be achieved when FAT activities are deferred to the site. Include a discussion of required reasoning/justification for deferring FAT activities and criteria which must be satisfied before FAT activity deferral can be performed and the post FAT activities that would have to be accomplished on site (versus the factory).

Westinghouse Response:

Per paragraph 7.3.1.5 the purpose of the FAT is to demonstrate that the complete system is integrated and functional. Further, it states that the FAT provides evidence to the customer that the system meets its requirements and provides confidence that the site installation and integration activities will be successful. These activities are the tests that show to the customer that the equipment is acceptable to transfer from the equipment vendor to the customer. As stated, deferring of FAT activities to site is based on customer agreement and is a contractual decision. From a technical perspective the ability to demonstrate acceptable integrated performance can be achieved in the factory or at site when it is integrated with site infrastructure. In this way the actual power grid, grounding plane, interconnecting cabling and other prototypic interface are available, thereby providing a more prototypic environment.

Per SPM paragraph 7.2.5.1 in the FAT paragraph “FAT is the equivalent of the description of Acceptance tests in IEEE Std. 1012.”

IEEE Std 1012-2004, “ acceptance testing: (A) Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system. (B) Formal testing conducted to enable a user, customer, or other authorized entity to determine whether to accept a system or component.”

The statement of deferring the testing to site does not eliminate the testing requirement of the equipment provider. Westinghouse experience is that customer and project schedules may benefit from delivering tested equipment and completing the final integration testing that represents part of the FAT at the site. An example or a condition where this may occur is when a system has completed system validation testing. In this testing the hardware was fully exercised but a future software baseline is planned due to identified design changes. The system can be delivered and installed and a follow on FAT test could be run at site at the new baseline. In this scenario the equipment vendor performs testing with the customer support to complete the intent of an acceptance tests. The controls and reporting of this type of scenario would be based on customer agreement and contractual obligation. The important aspect is that the correct amount of testing is performed on the system consistent with the test plan and the agreed acceptance criteria.

This allowance is to recognize the many combinations and conditions I&C systems are developed and delivered, such as entire systems, subsystems, back fits into existing systems, etc. The two aspects of a FAT are the scope of the testing that is the responsibility of the supplier and at what location the test is performed. By allowing testing

that is considered to be the responsibility of the supplier performed at the site recognizes that the scope of the test is as important as or more important than where it is conducted. Therefore, the SPM will be revised in Sections 7.2.4, 7.2.5.1, and 7.3.1.5 to clarify that a FAT is performed prior to the customer accepting the equipment or system:

Section 7.2.4, "Schedule," will be revised as follows:
"Factory Acceptance Test (FAT) – The FAT is to be executed on a deliverable system and must be completed and meet its approved requirements before ~~shipping the safety system to~~ the customer accepts the system. The FAT is typically performed in the factory but some portion of the test can be performed at site if agreed to with the customer. When performed on a deliverable system, the System Validation Test can fulfill the role of the Factory Acceptance Test."

Section 7.2.5.1, "Testing Hardware," will be revised as follows:
"Factory Acceptance Tests – These tests shall be conducted on the deliverable hardware assembled in cabinet(s) ~~for shipment to the customer~~ and configured with the application software. The integration and system validation test can be credited for applicable parts of the Factory Acceptance Test (FAT) when conducted on deliverable hardware. FAT is the equivalent of the description of Acceptance tests in IEEE Std. 1012 (Reference 8)."

Section 7.3.1.5, "Factory Acceptance Test (FAT)," will be revised as follows:
"The purpose of the FAT is to demonstrate that the complete system is integrated and functional. To this end, the optimum scenario is to perform this test in the manufacturing facility ~~with full interconnection of the deliverable system cabinets (across all divisions) and with application software~~. Prior to ~~shipment of equipment to the site~~acceptance of equipment by the customer, a Factory Acceptance Test (FAT) is performed as a manufacturing test to provide evidence to the customer that the system meets its requirements and provides confidence that the site installation and integration activities will be successful."

"FAT is performed to:
- Demonstrate that the system ~~being delivered~~ has been manufactured correctly and is acceptable to the customer"

6. **Integration Test Items**

The following Integration Test Items listed in Section 7.3.1.3, "Integration Test," have been removed from the SPM in Revision 5:

- Error Handling
- Communications
- Redundancy
- Diversity

Since the SPM no longer lists test items for integration, it is unclear to the NRC how the stated objectives for integration testing can be achieved. The NRC staff needs to understand why these test items were removed and how the objectives of integration testing will continue to be achieved in absence of these test items. Please provide additional information explaining removal of integration test items as well as justification for no longer performing these test activities as a part of integration testing.

Westinghouse Response:
Integration testing is defined as a functional test that is performed on a system that is now integrated. This integration is referring to the integration of released software with deliverable equipment or equivalent. Section 7.3.1.3 states, "Integration testing is used as part of system validation testing when validating the design and as part of the FAT testing to demonstrate the deliverable system has been properly integrated." Therefore, there are two types of integrated tests, System Validation Tests and FAT. For first of a kind testing, a System Validation Testing proves that the design meets detailed functional requirements and it demonstrates that the design is correct and adequate. The tests that you identified above have not been removed but are listed along with other functions in paragraph 7.3.1.4:

- Safety Functions
- Communications
- Displays
- Diagnostics
- Performance
- Error Handling – potential errors shall be handled with known consequences
- Communications – all defined outputs shall be broadcast and received correctly within the channel
- Redundancy – all shared inputs shall produce the same output from redundant processors
- Diversity – all functionally diverse signals shall be verified for correctness in termination

The Factory Acceptance Test (FAT) is also an Integration Test. It's performed to demonstrate that the delivered equipment has been manufactured correctly and integrated properly. The FAT also has a list of test that fulfills this definition as applicable for the system under test. These are listed in section 7.3.1.5:

The following test items shall be included or demonstrated in the FAT:
- Safety Functions
- Communications
- Operability of Displays
- Diagnostics associated with hardware specific inputs (door alarms, temperature alarms, breaker status, etc.)
- Performance (accuracy, time response, etc.)

As can be seen by this list, the FAT contains the tests from the previous list that are applicable to a FAT and the FAT is an extensive test of the integrated system and will demonstrate a proper operating system.

7. **Performance of FAT on Deliverable System**

SPM Section 7.3.1.5, "Factory Acceptance Test (FAT)," includes a description of the FAT which states that the FAT is to be executed on a deliverable system. The reworded description of FAT however seems to imply that some portion of the FAT may now be performed on a non-deliverable or surrogate system as follows:

FAT includes tests that are performed for each deliverable system.

Please confirm that FAT will not include tests that are performed on non-deliverable or surrogate equipment or provide a description and justification for crediting FATs performed on surrogate equipment to apply to deliverable systems.

Westinghouse Response:
It is agreed and confirmed that the Factory Acceptance Test (FAT) is performed on the deliverable equipment. The statement about surrogate equipment is referring to system validation testing and regression testing, which can be performed on surrogate equipment. This paragraph in section 7.3.1.5 is providing clarification for the scenario where a previous validation test has been performed on the first of a kind system but during the Nth of a kind, a design change has been identified. Such a change could be either hardware, software or both. A System Validation test would need to be run for the design change to prove that the design implementation is correct for all of the systems that are considered the same design, (i.e. the first of a kind and all Nth of kind systems). It is the System Validation test that can be run on surrogate equipment. Another option would be that the deliverable system that is going through the FAT test program can be the surrogate equipment for the purposes of System Validation testing for all other systems. Either way the system that is going through the FAT would need to obtain the change and appropriate FAT testing would be conducted for that deliverable system. Therefore, Section 7.3.1.5, "Factory Acceptance Test," will be revised as follows:

"~~As design changes are introduced, regression analysis shall be performed to determine what tests need to be repeated or introduced to maintain the level of system design validation achieved during the first of a kind system validation test program. The system validation tests required by the regression analysis may be performed on the deliverable equipment as a separate section of the FAT or on surrogate equipment consistent with the regression testing methods described in subsection 7.3.2.2.~~"

With that sentence moving to Section 7.3.1.4, "System Validation Test," as follows:
"As design changes are introduced, regression analysis needs to be performed to determine what tests need to be repeated or introduced to maintain the level of system validation achieved during the first of a kind test program. The system validation tests required by the regression analysis may be performed on the deliverable equipment as a separate section of the FAT or on surrogate equipment consistent with the regression testing methods described in subsection 7.3.2.2.

8. **Surrogate System Testing**

The revised test strategy outlined in the SPM includes provisions for using a test bed, proxy, or surrogate system in lieu of actual production equipment to be delivered to the site for performance of Integration and System Validation Tests. SPM, Section 7.3.1.5, "Factory Acceptance Test (FAT)," includes a description of the FAT which states that the FAT is to be executed on a deliverable system (i.e., not a surrogate system). However, Section 7.3.1.5 also states that System Validation Tests, which can be credited to fulfill the role of FAT, may be performed on surrogate equipment. These statements appear to contradict the purpose of the System Validation Test or the FAT and the conditions under which the testing is to be conducted (actual deliverable system versus surrogate system). Please clarify these statements and justify what specific conditions are appropriate to test a surrogate system, for either the FAT or System Validation Test rather than the production-based system.

Please provide additional information on the process for crediting system validation tests to meet FAT objectives. The NRC staff needs to understand any limitations or conditions for crediting System Validation Tests to meet FAT requirements before a safety determination can be made for this change.

Westinghouse Response:
For system validation test to be credited as FAT it must be performed on the delivered equipment.

In this version of the SPM, “Integration Test” is now a term that describes a condition of the test (e.g. integrated). There are two types of integrated testing that performs two different functions; System Validation Testing and Factory Acceptance Testing (FAT). The System Validation Testing is a test of the design (system design, Hardware design and the software design) that proves that the design as implemented meets the requirements. The FAT is a manufacturing integrated test that demonstrates that the deliverable equipment is working properly and consistent with the System Validation Test (e.g. within acceptance criteria). If the System Validation Test is performed on the deliverable system then it can also be credited as the FAT for that system. This has been the model for many plants where the System Validation Test was also the FAT. Once the design has been validated via the System Validation Test, it can now be credited for other systems of the same design.

Therefore, for every system delivered, either first of a kind or Nth of a kind (follow on units), we must show that it has passed a System Validation Test and a FAT. The System Validation Test can be performed on Surrogate equipment. This can be a test bed that is configured to be functionally equivalent to the production hardware or it could be production equipment destined to be delivered. Either way, it is considered to be surrogate equipment for follow on units.

The FAT is never performed on surrogate equipment as its purpose is to demonstrate acceptability of the delivered system.

Inherent in this strategy is that the FAT is a functional subset of the System Validation Test. For example, an analog input and accuracy test is performed as part of both tests. To prove the system is meeting all of its requirements in the design of hardware and software the system validation test checks every signal. And for the FAT each signal is also checked to confirm the correct manufacturing of every signal path. However, the detailed software that displays the information to the operator needs only be fully tested once during the System Validation Test and not during the FAT. The FAT needs to demonstrate the display is working and data communication to the display are working properly. Therefore, the FAT is a functional subset of the System Validation Test. The term functional subset is used because there may be a different and more efficient method of testing these features than to just rerun a subset of the System Validation Tests.

Therefore, every system delivered must show that it passed a FAT and a system validation test. And the system validation test may have been run on the delivered system in question, another delivered system of the same design or on other appropriate surrogate equipment. But all delivered systems must have a FAT run on that system. Therefore, Section 7.3.1.4, “System Validation Test,” will be revised as follows:

"See EXHIBIT 7-1 COMPARISON OF SYSTEM VALIDATION TEST AND FAT for a detailed description of the tests performed during system validation testing and FAT.

For system validation test to be credited as FAT, it must be performed on the delivered equipment.

As an alternative to functional testing with production hardware, a system validation test can be performed with a test bed…"

9. **Time Response Testing**

The Table in Exhibit 7-1, "Comparison of System Validation Test and FAT," includes a Test Item of "Performance" with a "Design Aspect" of "Time Response Testing." The corresponding System Validation Test and FAT items to demonstrate compliance refer to tests using representative functions and representative samples of tests instead of actual safety functions performed on production equipment. Please justify the use of representative tests and representative functions to assure compliance with time response requirements in lieu of testing actual functions using production equipment and the basis for doing so.

Westinghouse Response:
In this table the term representative means typical. It also means that it is not exhaustive for all combinations of every factor or path. Time response for a typical software based design has historically shown that the largest contributor to the variability of the response has been the software loop times and the asynchronous nature of signal propagation through such systems. During the System Validation Testing real trips and actuations are caused by the real inputs and the time is measured for multiple runs. These times are compared to the requirements and to the analyzed and predicted times to bound the response of the system. For the FAT, the design has been validated and the time response has been well characterized. Because these systems are highly digital, very little of the time response path are susceptible to latency issues that are not detectable during functional testing or identified as part of the system diagnostics. Therefore the FAT is intended to be a subset of the System Validation Testing but still tests the hardware paths or uses commercial dedication test data for time sensitive components. For example, signal conditioning front ends that have filters and latency limits can be better tested independently during the commercial dedication process on the bench. However, FAT time response testing exercises the actual safety function actuations and trips on the deliverable equipment. Therefore, the column, "FAT (Nth Application)" in Exhibit 7-1 will be revised as follows:

A representative sample of safety function tests on the deliverable hardware with the deliverable software to demonstrate critical safety trips, consistency with analytic model and first application response tests

- One path through ~~each relative~~critical hardware component; e.g., each PM, I/O module, high-speed datalink, etc.
- Component response confirmed by commercial grade dedication process (similar to spare parts).

10. **Archival Requirements - Section 4.11.2, "Archival Requirements"**

In Revision 4 of the SPM, the archival requirements are the responsibility of the software librarian and should be performed in accordance with Reference 4 (Westinghouse Level II Policies and Procedures). In Revision 5 of the SPM, the commitment is changed to, in part, "the requirements of this section 'can be' performed by the software librarian." Provide additional detail explaining what individual or group of individuals, by position, is (are) specifically responsible for completion of archival requirements associated with the development, control, storage, and distribution of all project software deliverable physical media.

Westinghouse Response:
Archival requirements are per the Westinghouse Level II Policies and Procedures. Ultimately the group managers are responsible for their group's work products being archived according to the procedures. Software Librarian is a role within the context of IV&V activities in addition to being a position within the IV&V Group. The activities of the role could be performed by individual other than the person whose title is Software Librarian. This provision was necessary to allow flexibility in task assignments within the IV&V group with discretion of the IV&V manager who ultimately is responsible for IV&V archival requirements.

11. **Independent Verification and Validation Organization – Section 2, "Organization"**

In Revision 4 of the SPM it was not permitted for IV&V team members to participate on the design team. In Revision 5 of the SPM, the requirement was relaxed such that only IV&V 'engineers' are not allowed to participate on design activities. Provide additional information related to the type of design activities and justification why some IV&V team members (i.e., not IV&V engineers) would be allowed to participate in design activities.

Westinghouse Response:
The IV&V Group consists of positions including engineers, administrative assistants, software librarians and escorts. The wording in this section was changed to 'engineers' to differentiate those resources who do perform design verification and validation activities. Individuals who are not performing design related IV&V functions can be shared by other organizations as their work scope are not related to 'design activities' and does not jeopardize independence. For instance, escorts (who are hired to escort foreign nationals and customers) can be loaned to other groups with no impact to IV&V work performed on any project. Therefore, Section 2, "Organization," will be modified as follows:

"Reference 11 requires that the IV&V team for a safety system is organized independently of the design team. The IV&V organization meets this requirement by not allowing IV&V ~~engineers~~ team members to participate on design activities, even on a part time basis, if they are involved in the verification of that design.

The IV&V Team in the context of this SPM refers to those individuals within the IV&V organization who perform V&V functions on the safety system design, implementation, and test (i.e. engineers and technicians). The IV&V organization may include other individuals who perform supporting roles that are not design verification related and the organizational independence does not apply to those individuals."

12. **Test Plans**

Section 3.3.5.7.1, “Test Plans,” of Revision 5 of the SPM describes that the test plan will contain the method for defining requirements to be tested and the method for establishing the acceptance criteria and how it will be documented. In Revision 4 of the SPM, the text stated, in part, “They [the test plans] shall contain all the requirements for all acceptance test procedures and define each required test to be conducted.” Please provide additional information explaining why it is acceptable to provide only a method for defining requirements and acceptance criteria rather than defining the actual test requirements and acceptance criteria as was previously required by the SPM and consistent with the definition and content of a “test plan” in accordance with IEEE Std. 829.

Westinghouse Response:
Per IEEE Std 829-1998 overview states:
“The test plan prescribes the scope, approach, resources, and schedule of the testing activities. It identifies the items to be tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with the plan.”

Additionally in section 4.2 on Test Plan the following is stated:

“If some or all of the content of a section is in another document, then a reference to that material may be listed in place of the corresponding content. The referenced material must be attached to the test plan or available to users of the plan.”

Additionally in section 4.2.3 on Test Items for the test plan the following is stated:

“Supply references to the following test item documentation, if it exists:

a) Requirements specification;”

This allows for a reference to the item if it exists. Review of IEEE 829-2008 also includes the allowance of references to requirements in paragraph “9.2.1 (LTP Section 2.1) Test items and their identifiers”

The reason for the change in the SPM is due to the typical sequence and progression of a project. Requirements analysis, testing coverage and tracing of the requirements to test cases are significant testing activities. The Test Plan is needed to outline these activities. The test planning and initial engineering work occurs in parallel with the finalization of the design requirements and the implementation specifications. Therefore, the specific requirements to be tested are not available or issued in their final form when the test plan is written. Our processes include an extensive requirements management process and analysis for coverage using appropriate tools. This includes linking to test procedures and/or test cases depending on the requirement level. The requirements that are being tested are tied to the test section or test procedure either directly or by reference to a requirements tracing document for testing.

Since IEEE 829 recognizes and allows the ability to provide a reference to the requirements to be tested, and the normal progression of a test program determines the

specific requirements and coverage at a lower level than what is available for the test plan, the SPM was changed to better reflect the typical process.

13. **Software V&V Plan Review**

Section 4.6.2.4, "Software Verification and Validation Plan Review," of Revision 5 of the SPM states, in part, "The SVVP (Section 5) *has been* reviewed for adequacy and completeness of the verification and validation methods for Common Q." In Revision 4 of the SPM it states, in part, that, "The SVVP *is* reviewed for adequacy and completeness of the verification and validation methods for Common Q." Why is it acceptable for the SVVP to no longer be reviewed for a new or ongoing project as part of the Westinghouse Global Management System Quality Procedures, the descendant of Reference 4 in Revision 4 of the SPM?

Westinghouse Response:

The SPM will be revised as follows (as originally stated in Revision 4):

"The SVVP (Section 5) ~~has been~~ is reviewed for adequacy and completeness of the verification and validation methods ~~for Common Q™~~ defined in the SVVP. An independent reviewer meeting the qualifications of Reference 4 performed this review as part of the review process for this SPM. Compliance to the SVVP is covered by the in-process audits described in subsection 4.6.2.7."

Section C

WCAP-16096-NP-A, Revision 5.1, “Software Program Manual for Common Q™ Systems,” (Non-Proprietary)

REVISION HISTORY

RECORD OF CHANGES

Revision	Revision Made By	Description	Date
0	Warren R. Odess-Gillett	See PRIME for description of change	May 2003
1	Mark J. Stofko	See PRIME for description of change	January 2004
1A	Mark J. Stofko	See PRIME for description of change	December 2004
2	Matthew A. Shakun	See PRIME for description of change	August 2010
3	Matthew A. Shakun	See PRIME for description of change	February 2012
4	Matthew A. Shakun	See PRIME for description of change	July 2012
4 (NRC Approved)	Matthew A. Shakun	See PRIME for description of change	February 2013
5	Matthew A. Shakun	See PRIME for description of change	August 2017
5 (NRC Approved)	Matthew A. Shakun	Revised to issue the NRC approved version. The SPM will also be revised to incorporate the changes as specified in LTR-NRC-18-36, “Responses to NRC Request for Additional Information for Westinghouse Topical Report WCAP-16096-P/WCAP-16096-NP, Revision 5, "Software Program Manual for Common Q™ Systems"	November 2018
5.1 (NRC Approved)	Matthew A. Shakun	Revised to add the following statement before NRC’s safety evaluation: “Westinghouse acquired the AC160 product line from ABB on April 29th, 2021. Any references in the following Safety Evaluation to ABB’s ownership of AC160 is historical in nature.” No changes were made to the SPM text. Change bars have been retained from the previous revision. See NA-SSPCE-21-0009-CQP-EC for change evaluation. See CQP-00136 for Request for Engineering Change. Note: The review performed by Richard M. Paese signifies the non-applicability of this topical report revision to the AP1000 project unless further AP1000 plant licensing action is taken.	See PRIME

TABLE OF CONTENTS

TABLE OF CONTENTS (cont.)

TABLE OF CONTENTS (cont.)

TABLE OF CONTENTS (cont.)

TABLE OF CONTENTS (cont.)

TABLE OF CONTENTS (cont.)

TABLE OF CONTENTS (cont.)

TABLE OF CONTENTS (cont.)

TABLE OF CONTENTS (cont.)

TABLE OF CONTENTS (cont.)

TABLE OF CONTENTS (cont.)

LIST OF TABLES

Table **Title** **Page**

LIST OF FIGURES

Figure **Title** **Page**

None.

ACRONYMS AND TRADEMARKS

The following abbreviations and acronyms are defined to allow an understanding of their use within this document.

Acronyms	Definition
ADR	Architecture Design Review
BTP	Branch Technical Position
CAPs	Westinghouse Corrective Actions Process
CCB	Configuration Control Board
CDA	Critical Digital Asset
CDR	Critical Design Review
CET	Core Exit Thermocouple
CGDP	Commercial Grade Dedication Program
COP	Continuity of Power
COTS	Commercial Off-The-Shelf
CPCS	Core Protection Calculator System
CS	Communication Section
DT	Design Team
EDMS	Electronic Document Management System
ELM	Engineering Line Manager
ENM	Existing Software not to be modified
EPM	Engineering Project Manager
ESFAS	Engineered Safety Features Actuation System
ETBM	Existing Software to be modified
FAT	Factory Acceptance Test
FCB	Function Chart Builder
FPDS	Flat Panel Display System
I&C	Instrumentation and Control
I/O	Input and Output
IEEE	Institute of Electrical and Electronics Engineers
IDI	Isolated Development Infrastructures
ILP	Integrated Logic Processor
ITP	Interface and Test Processor
IV&V	Independent Verification and Validation
HSI	Human System Interface
LCL	Local Coincidence Logic

ACRONYMS AND TRADEMARKS (cont.)

Acronyms	Definition
NA	Nuclear Automation
NPP	Nuclear Power Plant
NQA	Nuclear Quality Assurance
PAMS	Post Accident Monitoring System
PHA	Preliminary Hazards Analysis
PM	Processor Module
PPS	Plant Protection System
PQP	Project Quality Plan
PS	Processing Section
QMS	Quality Management System
RPS	Reactor Protection System
RTA	Requirements Traceability Analysis
RTM	Requirements Traceability Matrix
RVL	Reactor Vessel Level
SAT	Site Acceptance Test
SCA	Source Code Analyzer
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SCR	Software Change Request
SDD	Software Design Description
SHA	Software Hazards Analysis
SMP	Software Maintenance Plan
SPM	Software Program Manual
SQAP	Software Quality Assurance Plan
SRR	Software Requirements Review
SRS	Software Requirements Specification
SSP	Software Safety Plan
SVVP	Software Verification and Validation Plan
SVVR	Software Verification and Validation Report
SysRS	System Requirements Specification
USNRC	United States Nuclear Regulatory Commission
VT	Independent Verification and Validation Team

Autodesk and AutoCAD are registered trademarks of Autodesk, Inc.

ACRONYMS AND TRADEMARKS (cont.)

Microsoft®, Excel®, Windows® and Word® are registered trademarks of Microsoft Corporation in the United States and/or other countries.

IBM and Lotus Notes are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide.

All other product and corporate names used in this document may be trademarks or registered trademarks of other companies, and are used only for explanation and to the owners' benefit, without intent to infringe.

GLOSSARY OF TERMS

The following definitions are provided for the special terms used in this document. Definitions for all other terms used in this document can be found in Reference 5.

Term	Definitions
Channel	An arrangement of components and modules as required to generate a single protective action signal when required by a generating station condition. A channel loses its identity where single protective action signals are combined.
Configuration Status Accounting	The recording of information needed to effectively manage a software configuration.
Engineering Line Manager	The Engineering Line Manager (ELM) provides resource management of people and other resources (such as materials and equipment) to provide optimal implementation of customer projects for their assigned products and services.
Engineering Project Manager	The Engineering Project Manager (EPM) is assigned to a particular Common Q™ customer project and is responsible for the development, scheduling, financial and quality execution of the assigned project. The Common Q™ Platform Lead may be responsible for these functions for internal generic Common Q™ development activities. Organizationally, EPMs and Platform Leads directly report to an Engineering Line Manager (ELM). EPMs and Platform Leads may delegate the performance of necessary tasks to other persons but remain responsible for their execution.
Division	The designation applied to a given system or set of components that enables the establishment and maintenance of physical, electrical, and functional independence from other redundant sets of components.
Module	A module is the smallest software entity that is subjected to testing. It is a custom PC element or a type circuit in AC160 space or a subroutine in language programming space.

GLOSSARY OF TERMS (cont.)

Term	Definitions
Nuclear Automation	Nuclear Automation is the cognizant engineering organization within Westinghouse Electric Company that is responsible for the design and implementation of Common Q™ based systems.
Platform Lead	The Common Q™ Platform Lead is responsible for the platform development meeting the continuing needs of the product family.
Project Plan	A documented plan that identifies the information necessary to execute the project, such as: • Overview of Project/System • General Functions of the Software • Project scope • Deliverables • Project milestones • Project stages • Project inputs and review • Key personnel and project interfaces including — Internal — Customer — Supplier • Output review/verification/validation • Reference to detailed project schedule • Assumptions/Dependencies/Constraints/Risks • Methods, tools, and techniques • Performance measures • Security provisions • Software Lifecycle
Project Quality Plan (PQP)	A document that specifies alternatives or supplements to the Westinghouse QMS, Level 2, or Level 3 procedures as required to meet contractual requirements or quality standards other than those specified in the Westinghouse QMS. When the SPM refers to a PQP, it includes the Project Quality Plan and Project Plan (including the Software Development Plan) defined in the Westinghouse Quality Management System Procedures.

GLOSSARY OF TERMS (cont.)

Term	Definitions
Quality	"Quality" is the generic title of any of the independent Quality Assurance departments that are responsible for performing quality assurance functions. Each business unit has a quality organization that is separate from the engineering organization. The Quality organization provides oversight by way of periodic audits to verify that the Nuclear Automation organization is effectively implementing the Westinghouse Quality Management System and its implementing procedures.
RTA	The Requirements Traceability Analysis (RTA) is the task of ensuring the completeness and accuracy of the RTM; all lower level requirements and design features are derived from higher level requirements, and that all higher level requirements are allocated to lower requirements, design features, and tests. The traceability analysis also provides a method to cross-reference each software requirement against all of the documents and other software items in which it is addressed. The purpose of this analysis is to verify that the design team addresses every requirement throughout the design life cycle process. The IV&V team is responsible for performing the RTA.
RTM	The Requirements Traceability Matrix (RTM) is either a table of information prepared manually, or a report generated from a requirements database. The RTM associates requirements with the documentation and software that satisfies them. Requirements are entered in the matrix and are organized into successive lower level requirements as described in each document. The requirements are then traced through the software lifecycle to the design, code, and test documentation. The design team is responsible for creating the RTM to the point of identifying the code satisfying the requirement. IV&V will complete the RTM identifying validation of the requirement.
SAP	SAP is an enterprise software system used by Westinghouse Electric Company to support its business processes by providing an integrated data and process structure. It is provided by the German company "Systems, Applications and Products in Data Processing."

GLOSSARY OF TERMS (cont.)

Term	Definitions
Secure Development Environment	The condition of having appropriate physical, logical and programmatic controls during the system development phases (i.e., concepts, requirements, design, implementation, testing) to ensure that unwanted, unneeded and undocumented functionality (e.g., superfluous code) is not introduced into digital safety systems.
Secure Operational Environment	The condition of having appropriate physical, logical and administrative controls within a facility to ensure that the reliable operation of digital safety systems are not degraded by undesirable behavior of connected systems and events initiated by inadvertent access to the system.
Shall	When used in a sentence, “shall” denotes a required action.
Should	When used in a sentence, “should” denotes a recommended action.
Software Item	A software item is defined as collection of source code modules, object code modules, database modules, etc. which comprise the software running in one identifiable computer. Since a system may have multiple processors performing different functions, a system may have multiple software items.
System	A collection of components organized to accomplish a specific function or set of functions. Components may be hardware or software units.
Testing	The process of exercising or evaluating a system or system component by manual or automated means, to verify that it satisfies specified requirements or to identify differences between expected and actual results.
Unit (Software)	A unit consists of several modules that are integrated into a separately testable element, logically consistent with design specifications. It is a control module in AC160 space or a combination of modules in language programming space.

REFERENCES

Following is a list of references used throughout this document. Unless stated otherwise, the latest revision is applicable.

1. "Westinghouse Electric Company Quality Management System," Westinghouse Electric Company LLC.

2. ASME NQA-1-2008, Subpart 2.7, "Quality Assurance Requirements for Nuclear Facility Applications"

3. Guidance on Evaluation and Acceptance of Commercial Grade Digital Equipment for Nuclear Safety Applications, EPRI TR-106439, October 1996

4. Westinghouse Global Management System Quality Procedures

5. IEEE Std 610.12-1990 (Reaffirmed 2002), "IEEE Standard Glossary of Software Engineering Terminology."

6. IEEE Std 830-1998, "IEEE Recommended Practice for Software Requirements Specifications"

7. IEEE Std 1016-1998 (Reaffirmed 2009), "IEEE Recommended Practice for Software Design Descriptions"

8. IEEE Std 1012-2004, "IEEE Standard for Software Verification and Validation"

9. IEEE Std 1063-2001, "IEEE Standard for Software User Documentation"

10. IEEE Std 828-2005, "IEEE Standard for Software Configuration Management Plans"

11. IEEE Std 7-4.3.2-2003, "IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations"

12. IEEE Std 1008-1987 (Reaffirmed 2009), "IEEE Standard for Software Unit Testing"

REFERENCES (cont.)

13. IEEE Std 730-1998 “IEEE Standard for Software Quality Assurance Plans”

14. IEEE Std 829-1998, “IEEE Standard for Software Test Documentation”

15. IEEE Std 603-1991, “IEEE Standard Criteria for Safety Systems for Nuclear Power Generating Stations”

16. IEEE Std 1028-2008, “IEEE Standard For Software Reviews”

17. Reg. Guide 1.152, Rev. 3 (July 2011), “Criteria for Use of Computers in Safety Systems of Nuclear Power Plants”

18. Reg. Guide 1.168, Rev. 2 (July 2013), “Verification, Validation, Reviews, and Audits for Digital Computer Software used in Safety Systems of Nuclear Power Plants”

19. Reg. Guide 1.169, Rev. 1 (July 2013), “Configuration Management Plans for Digital Computer Software used in Safety Systems of Nuclear Power Plants”

20. Reg. Guide 1.170, Rev. 0 (Sept. 1997), “Software Test Documentation for Digital Computer Software used in Safety Systems of Nuclear Power Plants”

21. Reg. Guide 1.171, Rev. 1 (July 2013), “Software Unit Testing for Digital Computer Software used in Safety Systems of Nuclear Power Plants”

22. Reg. Guide 1.172, Rev. 1 (July 2013), “Software Requirements Specifications for Digital Computer Software used in Safety Systems of Nuclear Power Plants”

23. Reg. Guide 1.173, Rev. 1 (July 2013), “Developing Software Life Cycle Processes For Digital Computer Software Used In Safety Systems Of Nuclear Power Plants”

24. IEEE Std 1074-2006, “IEEE Standard for Developing a Software Project Life Cycle Process”

25. IEC-60880, Nuclear power plants – Instrumentation and control systems important to safety – Software aspects for computer-based systems performing category A functions – Edition 2.0

26. IEEE Std 1228-1994 (Reaffirmed 2010), “IEEE Standard For Software Safety Plans”

REFERENCES (cont.)

27. NUREG-0800, *Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants*, Chapter 7, "USNRC Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants," U.S. Nuclear Regulatory Commission, Revision 7, August 2016.

28. NUREG/CR-6430, "Software Safety Hazard Analysis"

29. ISO 90003, "Software engineering Guidelines for the application of ISO 9001:2008 to computer software"

30. (Reference Deleted)

31. (Reference Deleted)

32. NUREG/CR-6101, "Software Reliability and Safety in Nuclear Reactor Protection Systems," U.S. Nuclear Regulatory Commission

33. WCAP-17266-P, "Common Q Platform Generic Change Process," Westinghouse Electric Company LLC.

34. ASME NQA-1a-2009, "Addenda to ASME NQA-1-2008 Quality Assurance Requirements for Nuclear Facility Applications"

DOCUMENTATION REQUIREMENTS

Documentation requirements have been identified in this document. The following table identifies documents that are required by this SPM.

Table I. Document Requirements

Item	Title	Prepared By[1]
1	Audit Report (In-Process Audit)	Quality
2	Certificate of Conformance (System)	Quality
3	Code Review Report	Independent Reviewer from either the Design Team or IV&V Team
4	Coding Standards and Guidelines	Design Team
5	Commercial Grade Dedication Report	Design Team
6	Exception Report (Database)	IV&V Team or Design Team
7	Exception Report Log (Database)	IV&V Team or Design Team
8	Failure Modes and Effects Analysis	Design Team
9	Safety Classification Record	Design Team
10	Software Hazards Analysis Report	Design Team
11	Project Plan	Design Team
12	Project Quality Plan	Design Team
13	Project Schedule	Design Team
14	Purchase Order	Customer
15	Regression Analysis	Design Team or IV&V Team
16	Requirements Traceability Matrix (Database)	Design Team
17	Resource Plan	EPM
18	Software Change Request (Database)	Design Team or IV&V Team
19	Software Change Request Log (Database)	Design Team or IV&V Team
20	Software Design Description	Design Team

Table I. Document Requirements (cont.)

Item	Title	Prepared By[1]
21	Software Release Record	Design Team
22	Software Requirements Specification	Design Team
23	System Requirements Specification	Design Team
24	Technical Bulletin	ELM, EPM, or Platform Lead
25	Technical Manual	Design Team
26	Test Plan	IV&V Team or Design Team
27	Test Procedure (Module, Unit, Integration, System Validation, Factory Acceptance, Site Acceptance[2])	IV&V Team or Design Team
28	Test Report (Module, Unit, Integration, System Validation, Factory Acceptance, Site Acceptance[2])	IV&V Team or Design Team
29	Training Material	Design Team
30	Training Record	Each Employee
31	IV&V Report (Phase Summary/Final)	IV&V Team

Notes:

1. See Exhibit 5-1 for document preparation responsibilities.
2. Site testing may be performed by the Licensee, design authority, or site support group.

Information requirements have been identified in this document. The following table contains the section number where the requirement is identified, a description of the requirement, and the output document where the information should be located.

Table II. Information Requirements

SPM Section Number	Description of Requirement	Output Document
1.4.2	Training Record For SPM	Training Record
3.1.2	Defining Acceptable Risks	Project Plan
3.3.2	A detailed schedule	Project Plan

Table II. Information Requirements (cont.)

SPM Section Number	Description of Requirement	Output Document
3.3.2	Resource Plan	Project Plan
3.3.5.10	Software User Documentation	Technical Manual
3.3.5.11	Results of Software Safety Requirements Analysis	IV&V Report
3.3.5.12	Results of Software Safety Design Analysis	IV&V Report
3.3.5.13	Results of Software Safety Code Analysis	IV&V Report
3.3.5.14	Results of Software Safety Test Analysis	IV&V Report
3.3.5.15	Results of Software Safety Change Analysis	IV&V Report
3.3.6	Software Hazards	Software Hazards Analysis Report
3.3.6	Results of IV&V Analyses	IV&V Report
3.3.6	Information on suspected or confirmed safety problems	IV&V Report
3.3.6	Results of audits performed on software safety program tasks	Audit Report
3.3.6	Results of safety tests conducted on the system	Test Reports
3.3.6	Training Records	Training Record
3.3.6	Software Safety Certification – Code Certificate	IV&V Report
3.3.6	Tracking system to confirm hazards and their statuses are tracked throughout software life cycle	Requirements Traceability Matrix
3.3.10	Project Manager approves the use of any tool – approval implicit by listing tool in Plan	Project Plan
3.4.1	Software Hazards Analysis	Software Hazards Analysis Report

Table II. Information Requirements (cont.)

SPM Section Number	Description of Requirement	Output Document
3.4.2	Software Safety Requirements Analysis	IV&V Report
3.4.3	Software Safety Design Analysis	IV&V Report
3.4.4	Software Safety Code Analysis	IV&V Report
3.4.5	Software Integration Safety Analysis	IV&V Report
3.4.6	Software Safety Test Analysis	IV&V Report
3.4.7	Software Installation Safety Analysis	IV&V Report
3.4.8	Software Safety Change Analysis	IV&V Report
3.5	Training in SPM Section 11	Training Record
3.5.1	Review of Training Materials	IV&V Report
3.5.1	Personnel Training	Training Record
3.5.2.1	Review of Installation documentation	IV&V Report
3.5.2.2	Software Installation and Startup Procedure	Technical Manual
3.5.3	Procedures to verify software integrity to detect unauthorized modification of code or data	Technical Manual
4.1.1	Documenting Software Classification	Safety Classification Record
4.1.2	Commercial Grade Dedication	Commercial Grade Dedication Report
4.3.2.1	Quality Assurance Planning	Project Quality Plan
4.3.2.4	Verification of module code listings	Code Review Reports
4.3.2.6	Exception Report Log	Exception Report Database
4.3.2.6	Exception Report	Exception Report Database
4.5.1	Work Instructions	Any document required to supplement the SPM (such as Coding Standards and Guidelines Document)
4.5.2.1	Coding Standards	Coding Standards and Guidelines

Table II. Information Requirements (cont.)

SPM Section Number	Description of Requirement	Output Document
		Document
4.5.2.4	Metric Reporting	Test Reports
4.6.2.1	Software Requirements Review	IV&V Report
4.6.2.2.1	Architecture Design Review	IV&V Report
4.6.2.2.2	Critical Design Review	IV&V Report
4.6.2.3	Code Certification	Code Review Reports
4.6.2.4	SVVP Review	SPM
4.6.2.5	Functional Review	IV&V Report
4.6.2.6	Physical Review	IV&V Report
4.6.2.7	In-process Audits	Audit Report
4.6.2.8	Managerial Reviews	Audit Report
4.6.2.9	Software Configuration Management Plan Review	IV&V Report
4.6.2.10	Post Mortem Review	Corrective Action Report
5.1.4	Project-Specific IV&V Plan Activities	Project Plan
5.4.5.2	IV&V Checklists	IV&V Report
5.4.5.2	Review Changes to COTS software	Commercial Grade Dedication Report
5.4.5.3	Requirements Traceability Analysis	RTM or Requirements Management Database
5.4.5.4	Database reviews (see also 5.5.5.2 #5)	Implementation Phase Checklist in IV&V Report
5.5.1	Baseline Change Assessment	Regression Analysis
5.5.3.2	Software Safety Analyses	IV&V Report
5.5.4.2	Software Safety Design Analyses	IV&V Report

Table II. Information Requirements (cont.)

SPM Section Number	Description of Requirement	Output Document
5.5.5.2	Software Safety Code Analyses	IV&V Report
5.5.6	Software Safety Test Analysis	Test Phase Checklist in IV&V Report
5.5.6.3	Code Certificate	IV&V Report
5.5.7.1	Installation Procedures, System Generation Procedures, User Documentation	Technical Manual
5.5.7.2	Training Material	Training Program Per Customer Requirements
5.5.8	Regression Analysis	IV&V Report or separately prepared document
5.6.1	Discrepancy Reports	Exception Record Database; Status defined in IV&V Report
6.2.2.1	Identify original software items developed under this SPM for generic application that are to be controlled via SCM	Project Quality Plan
6.2.2.3	Define software items which are to be controlled via SCM	Project Plan
6.3.2	Master list of software under configuration control for a project	Configuration Management Release Report
6.3.2	Software Change Request	Database
6.3.2	Software Change Request Log	Database
6.3.3	Configuration Status Accounting	Configuration Management Release Report
9.3.2.1	Feasibility Analysis	Project Quality Plan
9.3.2.2	Detailed Analysis	SysRS, SRS, Test Plan, PQP
9.5.2.4	Risk Analysis	Project Quality Plan

Table II. Information Requirements (cont.)

SPM Section Number	Description of Requirement	Output Document
11.2	Justification for not performing complete system retesting	Regression Analysis in Exception Report or SCR
11.2	Exception Reports	Database

(Last Page of Front Matter)

SECTION 1
INTRODUCTION

1.1 PURPOSE

Computer software is essential to the design, analysis, operation and control of Common Qualified (Q) systems. This Software Program Manual (SPM) describes the requirements for the software design and development process including the software/hardware interface. The SPM also describes the requirements for the use of software in Common Q™ systems. The SPM expands the procedural requirements for computer software in the Westinghouse Level II Policies and Procedures (Reference 4). This manual is compliant with (ASME) NQA-1-2008 (Reference 2), Subpart 2.7, (ASME) NQA-1a-2009 (Reference 34), and ISO 90003 (Reference 29).

The Requirements for the Common Q™ hardware design process are defined in Reference 4. Hardware verification is performed as part of the hardware quality assurance activities that are also defined in Reference 4.

The Software Program Manual consists of several basic elements:

1. A **Software Safety Plan**, which identifies the processes that, will reasonably assure that safety-critical software does not have hazards that could jeopardize the health and safety of the public.

2. A **Software Quality Assurance Plan** (SQAP), which describes the process and practice of developing and using software. The SQAP addresses standards, conventions, reviews, exception reporting and other software quality issues.

3. A **Software Verification and Validation Plan**, which describes the method of assuring correctness of the software.

4. A **Software Configuration Management Plan**, which describes the method of maintaining the software in an identifiable state at all times.

5. A **Software Test Plan**, which describes the method for testing software.

6. A **Software Installation Plan**, which describes the method for installing software.

7. A **Software Maintenance Plan**, which describes software practices after delivery to a customer.

8. A **Secure Development and Operational Environment Plan**, which provides reasonable assurance that Common Q™ Systems and the development environments in which they are created are protected from inadvertent operator actions and undesirable behavior of connected systems.

The SPM also discusses Software Management, documentation and other matters related to software design and use.

It is intended that this SPM be consistent with NRC regulatory positions taken with respect to specific IEEE standards. These regulatory positions are documented in the Standard Review Plan (NUREG-0800) and its associated Branch Technical Positions and Regulatory Guides.

EXHIBIT 1-1 RELATIONSHIP OF SPM TO IEEE STANDARDS shows how the IEEE standards are applied to various Common Q™ design and quality assurance activities. The block labeled *System* depicts IEEE Standard 603 (Reference 15) and IEEE Standard 7-4.3.2 (Reference 11), that support systems development. The former addresses computer and non-computer hardware elements while the latter addresses system-level issues for software. The block labeled *Design Output Activities* shows the various software design activities and the specific IEEE standards that support those activities. IEEE Std 1074 (Reference 24) as endorsed by RG 1.173 (Reference 23) addresses the development of software life cycle processes, and therefore serves to unify the individual activity standards. It also addresses assurance activities, referred to by IEEE Std 1074 as "integral processes." These are shown on the bottom of the exhibit.

1.2 SCOPE

1.2.1 Software Classification and Categorization

This SPM shall apply to all software and firmware, whether developed in-house, licensed or procured from a commercial vendor, obtained from another organization or otherwise acquired and used in a Common Q™ system for delivery to a customer.

The Common Q™ software systems and software modules are identified as belonging to one of the following classes:

- **Protection** (safety critical – critical performance of the system). Software whose function is necessary to directly perform RPS control actions, ESFAS control actions, and safe shutdown control actions (Meets 10 CFR 50 Appendix B requirements).

- **Important-to-Safety** (important system performance). Software whose function is necessary to directly perform alternate protection system control actions or software that is relied on to

monitor or test protection functions, or software that monitors plant critical safety functions (Meets 10 CFR 50 Appendix B requirements).

- **Important-to-Availability**. Software that is relied on to maintain operation of plant systems and equipment that are critical to maintaining an operating plant.

- **General Purpose**. Software that performs some purpose other than that described in the previous classifications. This software includes tools that are used to develop software in the other classifications, but is not installed in the online plant system. Examples of commercially dedicated General Purpose software include compilers, assemblers, linkers, comparators, and editors. Examples of Westinghouse developed General Purpose software include test case generators, and test tools (e.g., I/O Simulator).

The requirements for the classification of functions, systems and equipment are provided in Reference 4. The classifications of the system functional level are shown in EXHIBIT 4-1 ASSIGNMENT OF COMMON Q™ SOFTWARE TO CLASSES.

The SPM makes distinctions regarding the methods applied to each of the above classes. Specific parts of the software in a single system may be assigned to different classes. Each part of the software must have an assigned class. Common Q™ applications not listed in EXHIBIT 4-1 ASSIGNMENT OF COMMON Q™ SOFTWARE TO CLASSES shall document the software classification using the Safety Classification Record in accordance with the requirements of Reference 4.

The SPM makes distinctions regarding methods applied to each of the following categories of Common Q™ software:

- Original, Developed for a Common Q™ System
- Existing, to be Modified
- Existing, to be used as is

Software in several categories may be included in each Common Q™ system. For example, a typical computer system may rely on:

- An operating system from a commercial supplier that is existing, used as is.
- Some residual code to be updated from a previous project (existing, to be modified)
- New algorithms (originally developed)

This SPM applies to all software used in the development, testing or delivered Common Q™ systems.

1.2.2 Software Exclusions

The following software is excluded from the requirements of this SPM:

- Administrative software used for purposes such as ordering, scheduling and project management.
- Commercial applications software for use in database management systems, word processing, and commercially purchased CAD systems. Such applications are Microsoft® Excel®, Microsoft Word®, and AutoCAD® software.

1.3 OVERVIEW

Common Q™ software developers shall proceed through a software development effort by following the approach described in this manual.

The Software developers shall first become familiar with the Software Quality Assurance Plan (SQAP, Section 4). All activities relating to Common Q™ software development and maintenance shall be performed in accordance with the requirements contained in the SQAP.

The Engineering Project Manager (EPM) is required to determine the class and category of all software to be used for the Common Q™ system as described in the SQAP. The EPM is also required to identify the applicable standards that must be followed for those specific classes and categories of software. This information shall be documented. The software tasks and responsibilities are outlined in the SQAP based upon software classification and category.

Each quality assurance task is described in the SQAP for each software life cycle phase. The narrative description, along with the corresponding Exhibit, assist the EPM in making the required decisions concerning the appropriate tasks to be performed and who is responsible for performing them. In addition, the specific documents that must be produced for each software life cycle phase are discussed in the SQAP. Required documents vary for each software category.

The Software Verification and Validation Plan and the Software Configuration Management Plan describe the details of some of the activities outlined in the SQAP.

Adherence to the Software Verification and Validation Plan (Section 5) will verify the accurate translation from one step in the software development process to the next step and the validation that the software product fulfills the requirements for which the software was developed. The degree of independence required by this plan varies with the software classification. The applicability of the tasks varies with the software category. The general definition of and qualifications for reviewer independence are stated in Reference 4.

The Software Configuration Management Plan (Section 6) describes the procedures necessary to maintain the Common Q™ software in an identifiable state at all times. These procedures do not vary with the software class or category.

The Software Test Plan (Section 7) describes the testing process for Common Q™ safety systems. This plan identifies testing activities and test documentation required to verify and validate a Common Q™ safety system throughout the software life cycle.

The Software Installation Plan (Section 8) describes the method for installing operating system software and application software onto the processor module, and the method for installing operating system software and application software into the Flat Panel Display System.

The Software Maintenance Plan (Section 9) describes the activities necessary to maintain the Common Q™ software, to remove errors, to respond to new or revised requirements and to adapt the software to changes in operating environments.

The Documentation section (Section 10) of this Software Program Manual describes the various documents that are required. The set of required documents for each software class is specified in the Software Quality Assurance Plan.

The Problem Reporting and Corrective Action section (Section 11) of the Software Program Manual describes procedures necessary to track that all software errors and failures are promptly acted upon and in a uniform manner encompassing all Common Q™ software. The procedures in this section tie together the requirements of the Software Verification and Validation Plan and the Software Configuration Management Plan.

The Secure Development and Operational Environment Plan (Section 12) provides reasonable assurance that Common Q™ Systems and the development environments in which they are created are protected from inadvertent operator actions and undesirable behavior of connected systems.

1.4 GENERAL REQUIREMENTS

The management and control of software applies to computer software and associated documentation developed or used for Common Q™ applications. Software shall be developed, acquired, procured, controlled, and maintained in accordance with this Software Program Manual. Any software developed under a different program than this SPM will go through a Commercial Grade Dedication process, which evaluates the development of that software to the requirements of the SPM. A Commercial Grade Dedication Report will be produced for this software. The SPM meets the requirements of Reference 11 as augmented by Reference 17 for Protection class software. See EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES for the requisite activities for Important-to-Safety class software.

1.4.1 Software Life Cycle

The Software Life Cycle used in this SPM is based on information contained in References 2 and 24. The Software Life Cycle phases are:

- Concept
- Requirements Analysis
- Design
- Implementation or Coding
- Test
- Installation and Checkout
- Operation and Maintenance
- Retirement

These phases may overlap or be performed iteratively. If the phases overlap, each phase shall be completed before any subsequent phase is completed.

1.4.2 Indoctrination and Training

Personnel involved in Common Q™ software design and development shall have documented training in this SPM. Such training records shall be prepared and maintained in accordance with the requirements of Reference 4.

(Last Page of Section 1)

SECTION 2
ORGANIZATION

Reference 1 defines the Westinghouse Quality Policy, which is to provide products and services that fully satisfy customer and regulatory requirements. The Westinghouse President and CEO defines the overall quality policy and promotes a culture of conformance to requirements, customer satisfaction and continual improvement. Organizations reporting to the Westinghouse President and CEO are assigned responsibilities for contractual requirements being identified and met, a focal point for achieving customer satisfaction, and the quality of items and services. These organizations include functions such as Engineering, Manufacturing, Project Management, Quality, Marketing, and Purchasing. Reference 1 provides typical operational organization reporting structures designed to satisfy the commitments of the Quality Management System.

The methodology and procedures described in this SPM are implemented by the Nuclear Automation (NA) organization. Within this organization, software activities are organized into the following two teams:

- The **Design team** performs software configuration management activities, develops the system requirements, software design, and code for the Common Q™ systems. The design team may also develop common software that is used in systems developed by other groups.

- The **IV&V team** performs software design verification, software validation testing on the Common Q™ systems. Depending on the software classification, the design team may perform the validation testing activities.

Reference 11 requires that the IV&V team for a safety system is organized independently of the design team. The IV&V organization meets this requirement by not allowing IV&V team members to participate on design activities, even on a part time basis, if they are involved in the verification of that design. EXHIBIT 2-1 DESIGN/IV&V TEAM ORGANIZATION shows the relationship between the design team and the IV&V team. The IV&V team reports to an Engineering Line Manager (ELM) who is administratively and financially independent from the design team manager.

The IV&V Team in the context of this SPM refers to those individuals within the IV&V organization who perform V&V functions on the safety system design, implementation, and test (i.e. engineers and technicians). The IV&V organization may include other individuals who perform supporting roles that are not design verification related and the organizational independence does not apply to those individuals.

Team leaders are assigned specific responsibilities and the authority to assure the accomplishment of software management and control through written plans, procedures, standards, and instructions.

The Engineering Project Manager (EPM) is the manager of the group responsible for control of a software configuration item. The EPM may delegate the performance of necessary tasks to other persons but remains responsible for their execution. The EPM is ultimately responsible and accountable for:

- Plans, schedules, procedures, methods, and techniques required in the technical and administrative performance of the Common Q™ related software.
- Compliance with this Software Program Manual.

The overall effectiveness of the implementation of the SPM is evaluated by the Westinghouse Quality organization in accordance with the internal audit requirements of Reference 4.

(Last Page of Section 2)

SECTION 3
SOFTWARE SAFETY PLAN

3.1 INTRODUCTION

3.1.1 PURPOSE

The goal of this safety plan is to enable the development of safety critical software for Common Q™ Systems that has reasonable assurance that software defects do not present severe consequences to public health and safety.

3.1.2 SCOPE

The safety objective of this plan is to provide procedures and methodologies for the development, procurement, maintenance, and retirement processes of Common Q™ safety critical software to mitigate the potential of a software defect jeopardizing the health and safety of the public.

Any acceptable risks and safety objectives specific to a project shall be defined in the specific Project Plan for a given system implementation.

This plan is prepared in accordance with Reference 27, Branch Technical Position (BTP) 7-14, "Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems", and Reference 32. It applies to all Common Q™ safety critical software whose failure could result in severe consequences to public health and safety. For Common Q™ systems, safety critical software is defined as software belonging to the “Protection” class as defined in Section 1.

3.2 DEFINITIONS, ACRONYMS, ABBREVIATIONS, AND REFERENCES

Refer to page xiii for a list of acronyms and trademarks. Refer to page xvi for definitions. Refer to page xx for a list of references. This SPM itself complies with all referenced standards unless otherwise stated herein. As a result it will not be necessary for Westinghouse to show further compliance on individual projects unless otherwise defined as part of the project’s contractual requirements.

3.3 SOFTWARE SAFETY MANAGEMENT

In compliance with Reference 32, this section provides a description of the software safety organization, and the management of software safety activities and safety analysis requirements.

3.3.1 Organization and Responsibilities

Section 2 defines the organization that is responsible for design and implementation of Common Q™ Protection software.

The software safety organization is composed of three parts:

1. The Quality organization, an independent quality assurance department, coordinates and reviews quality assurance procedures and directives. The Quality organization has a reporting chain separate from the design team such that the QA organization is independent of project schedule and cost considerations. The Quality organization provides oversight by way of periodic audits to verify that the NA organization is correctly abiding by both the procedures and directives generated by both organizations. The Manager of the Quality organization, or designee, approves this Software Program Manual which includes the Software Safety Plan.

2. An independent V&V team within the NA organization performs the safety activities for a given Common Q™ system implementation project. Refer to subsection 5.4.3.2 for a description of the IV&V team.

3. An Engineering design organization with a VP in which an Engineering Director reports that is responsible for the design work. The design team ELM reports directly to that Engineering Director.

The IV&V ELM shall have the following software safety responsibilities:

1. Confirm there is sufficient, independent, technically qualified and trained resources to implement the requirements of this software safety plan. Training includes familiarizing the IV&V team members with the methods, tools and techniques described in subsection 5.4.5.

2. Coordinate software safety task planning and implementation with the design team for the activities in Section 5.

3. Verify that records are kept in accordance with Section 5 and Reference 4.

4. Support the QA department on any audits within the purview of its responsibilities.

The mechanism for communicating safety concerns, raised by project staff, to software safety personnel is defined in Reference 4.

3.3.2 Resources

NA management shall develop an early understanding of the resources required to develop Protection class software, so that these resources are put in place when they are required. The EPM and the IV&V team leader shall determine the resources required to implement a Common Q™ system. ELMs shall assign the appropriate resources to the EPM and IV&V team leader. The following resources are considered for both the design and IV&V team:

- Personnel
- Test materials and data
- Computers and other equipment
- Equipment support
- Tools
- Financial and schedule

The EPM shall maintain an up-to-date resource plan and assure that the resources are made available when required.

Project schedules and resource allocations are established via the Project Plan.

3.3.3 Staff Qualifications And Training

The qualifications and training requirements for those personnel performing software safety functions are primarily the same as those for performing the software design.

The following table identifies the personnel that will perform the tasks identified in Reference 32, subsection 3.1.5:

Table 3.3.3-1. Software Safety Task Assignments

Task	Assignee
Define safety requirements	Design Team
Design and implement safety-critical portions of the system	Design Team
Perform software safety analysis tasks	IV&V Team
Test safety-critical features	IV&V Team
Audit software safety plan implementation	Quality organization
Perform process certification	Quality organization (subsection 3.3.13)

One of the most important factors in developing reliable software is the development and use of a qualified staff. In assessing the training requirements, the Engineering Line Manager considers that:

- Training needs vary by individual
- Training and retraining may be needed at various project phases
- Staff qualification and training need to be periodically reassessed

In addition to the above, the IV&V team shall be trained in the tools, techniques, and methodologies described in subsection 5.4.5.

The ELMs assure that all personnel participating in the design, implementation, test and verification of software are qualified to perform their assigned tasks. Since there is currently no industry sanctioned certification program for Protection and Important-to-Safety class software personnel, the ELM assesses the capabilities of candidates and selects appropriately qualified personnel based on the manager's experience.

In determining whether any candidate is qualified, the ELM considers whether the candidate:

- Understands the system and its potentially hazardous effects, as described in Section 3.4
- Understands the job to be performed
- Has, or is capable of obtaining, working knowledge of system software and tools required to do the job
- Possesses the combination of skills and knowledge to perform the job; through a proper level of formal education, supplemental training, and experience
- Understands the related quality assurance, configuration management, and verification and validation plans
- Is able to produce reliable software, good documentation, and can implement required quality assurance practices

Throughout a project, requirements and tasks may change. The ELM shall periodically reassess the qualifications of all personnel working on Protection class software, particularly when specific changes to the project become known. The ELM may direct additional training before the changes are effective, in order to staff a fully qualified project team.

Personnel performing software safety reviews shall meet the qualifications for an independent reviewer, as defined in Reference 4.

3.3.4 Software Life Cycle

The software life cycle to be implemented for Common Q™ system development activities including IV&V is defined in subsection 1.4.1. Section 3.4 describes the relationship among specific software safety analysis tasks and the associated activities for each phase of the software life cycle.

3.3.5 Documentation Requirements

The documentation for Common Q™ software shall be prepared in accordance with the requirements in Section 10, and incorporates the software safety documentation requirements. The change and approval process for the Protection class software portions of project documentation is the same as for other documentation as specified in Section 4.6.

3.3.5.1 Software Project Management

A Project Quality Plan, compliant with Reference 4 shall be developed that will coordinate both the system development, software safety and quality assurance activities to identify the prescribed procedures and provide the adequate, allocated resources for their proper execution.

3.3.5.2 Software Configuration Management

Section 6 contains the requirements for software configuration management. Any deviations to these requirements shall be documented in the project specific Project Quality Plan. Section 6 defines specific SCM responsibilities for a Common Q™ project and covers each phase of the software life cycle.

3.3.5.3 Software Quality Assurance

Section 4 is the Software Quality Assurance Plan (SQAP) that describes the requirements and methodology to be followed in developing, acquiring, using and maintaining safety-critical software. This SQAP is compliant to Reference 13.

3.3.5.4 Software Safety Requirements

The system requirements documentation (Section 10.2) specifies the safety requirements to be met by the software to avoid or control system hazards.

3.3.5.5 Software Design Description

The Software Design Description (SDD, Section 10.3) includes descriptions of the software design elements that satisfy the software safety requirements.

3.3.5.6 Software Development Methodology, Standards, Practices, Metrics and Conventions

The standards, practices, conventions and metrics to be applied to the Common Q™ project are defined in Section 4.5.

3.3.5.7 Test Documentation

Test documentation includes test plans, test procedures and test reports. Test procedures incorporate test design and test cases.

3.3.5.7.1 Test Plans

The test plans provide a high level description of tests that will be conducted for the Common Q™ project. The plan will contain the method for defining the requirements to be tested, the method for establishing the acceptance criteria and how it will be documented. It also defines the methodology for the disposition of test exceptions (errors). This document is verified against the outputs generated from the requirements phase of IV&V for completeness. All prerequisites for testing shall also be identified in the detailed test sections. Subsection 4.3.2.2 describes the requirements for a test plan.

3.3.5.7.2 Test Procedures

The test procedures are the instructions for the actual tests conducted on the Common Q™ software. They include test setup, precautions and limitations, prerequisites, and the test cases used to validate proper operation. The test procedures are verified against both the test plan and outputs generated from the requirements phase of IV&V. Refer to Section 5.8.2 for a description of test procedure contents.

3.3.5.7.3 Test Reports

The test reports document the execution of the test procedures. In addition to attaching the signed and checked off test results, the test reports provide an overall summary of the test results and the resulting Exception Reports generated during the test. The system configuration at the time of test execution is also documented in the test reports. Test Reports are prepared in accordance with Section 5.8.3.

3.3.5.8 Software Verification and Validation

The Software IV&V documentation is described in Section 5.

3.3.5.9 Reporting Safety Verification and Validation

IV&V reporting is described in Section 5.

3.3.5.10 Software User Documentation

User documentation is described in Section 10.6.

3.3.5.11 Results of Software Safety Requirements Analysis

The results of the Software Safety Requirements Analysis as described in subsection 3.4.2 below shall be documented in the Requirements Phase section of the IV&V Report (Section 10.5).

3.3.5.12 Results of Software Safety Design Analysis

The results of the Software Safety Design Analysis as described in subsection 3.4.3 below shall be documented in the Design Phase section of the IV&V Report (Section 10.5).

3.3.5.13 Results of Software Safety Code Analysis

The results of the Software Safety Code Analysis as defined in subsection 3.4.4 below shall be found in the IV&V Report for the Implementation Phase of the software life cycle. Any changes will be documented in either IV&V Discrepancy Reports or as suggestions in the IV&V Report.

3.3.5.14 Results of Software Safety Test Analysis

The results of the Software Safety Test Analysis as defined in subsection 3.4.6 below shall be found in the IV&V Report for the Testing Phase of the software life cycle.

3.3.5.15 Results of Software Safety Change Analysis

The results of the Software Safety Change Analysis as defined in subsection 3.4.8 below shall be found in the IV&V Report. For each software life cycle that is revisited by the design team, the IV&V team will analyze the impact on the previous life cycle phase as well as the phase it is analyzing. The results of each phase's analysis will be found in the IV&V Report for that software life cycle phase.

3.3.6 Software Safety Program Records

Records generation and maintenance procedures required for Common Q™ software are described throughout this Software Program Manual. Originals of issued documents for Common Q™ software are maintained according to Section 10.

Before the software requirements phase is completed and after the overall system design is known, an evaluation is made to determine the safety critical hazards posed by the system through its interfaces. The analysis assumes that a worst case scenario of possible errors (hardware or software) has occurred in the system. Based on this assumption, the analysis results in an identification of system malfunctions that are injurious to public health and safety.

For each hazard identified above, the analysis further determines whether a software malfunction could produce the hazardous condition. These software hazards are identified in the Software Hazards Analysis Report as described in subsection 3.4.1. Each software producible hazard is evaluated during each phase of development of the safety critical software. The Software Hazards Analysis Report is issued by the Design Team and is an input to the IV&V team.

Results of IV&V analyses performed on requirements, design, code, test and other technical documentation are documented in the IV&V Phase Summary Reports and the Final IV&V Report. Information on suspected or confirmed safety problems in the prerelease or installed system is recorded in the Final IV&V Report. Results of audits performed on software safety program tasks are documented in the Quality organization's Audit Report. Results of safety tests conducted on all or any part of the entire system are documented in the Test Report. Training records are maintained by NA line management per Reference 4. Software safety certification is documented in the Code Certificate.

Retention of software safety program records is in accordance with Reference 4. The initiation and completion criteria for software safety program tasks for each phase in the software life cycle are defined in Section 5.

The tracking system used to confirm that hazards and their status are tracked throughout the software life cycle through retirement is the RTA and RTM as described in Section 5.

3.3.7 Software Configuration Management Activities

A key factor in developing reliable software is strict and detailed configuration management. Software configuration management activities for Common Q™ software are described in Section 6.

3.3.8 Software Quality Assurance Activities

Software quality assurance activities for Common Q™ software are described in Section 4.

3.3.9 Software Verification and Validation Activities

Software verification and validation activities for Common Q™ software are described in Section 5. These activities conform to the requirements in References 8 and 11.

3.3.10 Tool Support and Approval

Section 4.9 describes the use of software tools that are used in development of Common Q™ systems. Tools may produce better program structure and more reliable software through the automation of repetitive or time-consuming tasks. The EPM and IV&V team leader approve the use of any tool. This approval is based on an evaluation of the tool's readiness for use on a project involving Protection class software. This evaluation considers:

- The tool's past performance
- The extent of tool validation already performed
- The consistency of tool design with planned use
- The use of tool upgrades
- The retirement of tools
- The restrictions on the use of the tool due to limitations

The inadvertent introduction of software hazards by project tools is mitigated by the proper use of techniques for software configuration management, software quality assurance and IV&V as described in this SPM.

3.3.11 Previously Developed or Purchased Software

Subsections 4.1.2, 4.12.1, and 5.5.3.2 describe the requirements for using existing software, including purchased software, as safety critical software. WCAP-17266-P, "Common Q Platform Generic Change Process," (Reference 33) describes the change analysis for previously developed software to preserve the safety integrity.

3.3.12 Subcontract Management

Subsection 4.12.2 specifies the provisions for ensuring that subcontractor software meets established software safety program requirements.

3.3.13 Process Certification

An audit report from an In-Process Audit described in subsection 4.6.2.7 is prepared by the Quality organization to document that the software related activities were performed in accordance with the Quality Management System (Reference 1) and its implementing procedures.

3.4 SOFTWARE SAFETY ANALYSES

3.4.1 Software Safety Analyses Preparation

It is vitally important to understand the ways that a system could potentially present hazards to public health and safety. The system design and review techniques described in this SPM are used to avoid, preclude, or mitigate the impact of potential software hazards in systems built using the Common Q™ platform. Systems that include both Protection and Important-to-Safety class software need to postulate in the Software Hazards Analysis potential software hazards in the Important-to-Safety class software and the impact on Protection class software.

A Software Hazards Analysis (SHA) will identify the following:

- **Hazardous System States**. Before the software requirements phase is completed and after the overall system design is known, an evaluation is made to determine the safety hazards posed by the system through its interfaces that are injurious to public health and safety. The plant safety analysis defines the safety-critical hazards (accidents) posed by the plant that may be injurious to public health and safety. The failure modes and effects analysis performed for the specific Common Q™ System analyzes the vulnerability to single failures at the hardware module level, including existing compensating provisions (hazard controls) within the design of each system. These two sources form the design bases for software safety requirements for the Common Q™ Safety System.

- **Sequences of actions that can cause the system to enter a hazardous state**. For each identified hazard, the analysis determines whether a software malfunction could produce the hazardous condition, or the hazard could affect software operability. These hazards are identified in the Software Hazards Analysis Report. Each software related hazard is evaluated during each phase of development of the Protection class software. Reference 28 shall be used as a guide in performing this analysis.

- **Sequences of actions intended to return the system from a hazardous state to a non-hazardous state**. For each hazardous state, the system design must account for returning the system to a non-hazardous state. In preparing the Software Requirements Specification, the software developer considers techniques that can avoid a hazardous condition, or return the

system to a non-hazardous state. The result of the requirements phase may be a set of required or forbidden design, coding or testing techniques. The requirements phase may also identify specific tests to be performed or the implementation of certain hazard recovery techniques.

The System Requirements Specification (subsection 10.2.1) provides the high-level system design as required in subsection 4.4.1 b) of Reference 26. The interfaces between the software and the rest of the system are defined in the Software Requirements Specification (subsection 10.2.2).

3.4.2 Software Safety Requirements Analysis

In preparing the Software Requirements, the software developer considers techniques that can avoid a hazardous condition. The result of the requirements phase may be a set of required or forbidden design, coding or testing techniques. The requirements phase may also identify specific tests to be performed or the implementation of certain hazard recovery techniques.

Refer to subsection 5.5.3 for a description of the software safety requirements analyses performed. These activities provide reasonable assurance that each system safety requirement is satisfied by the software safety requirements.

3.4.3 Software Safety Design Analysis

Refer to subsection 5.5.4 for a description of the software safety design analyses performed. These activities provide reasonable assurance that each software safety requirement is satisfied by the software safety design.

3.4.4 Software Safety Code Analysis

Refer to subsection 5.5.5 for a description of the software safety code analyses performed. These activities provide reasonable assurance that each software safety design element is satisfied by the software safety code.

3.4.5 Software Integration Safety Analysis

The software integration safety analysis is performed as part of the software safety test analysis. Refer to subsection 3.4.6 for the software safety test analysis.

3.4.6 Software Safety Test Analysis

Refer to subsection 5.5.6 for a description of the software safety test analyses performed for system level testing. These activities provide reasonable assurance that each system and software safety requirement is

tested. Module/unit testing are included as part of Software Safety Code Analysis as described in subsection 5.5.5.

3.4.7 Software Installation Safety Analysis

Subsection 5.5.7 fulfills the requirements for a software installation safety analysis. This final safety analysis verifies that the installed system operates correctly.

3.4.8 Software Safety Change Analysis

Subsection 5.5.8 and Section 9 fulfill the requirements for a software safety change analysis. These activities provide reasonable assurance that changes to safety critical software do not create, impact a previously resolved, or exacerbate a currently existing hazard, and does not adversely affect any safety-critical software design elements.

3.5 POST DEVELOPMENT

In spite of the best efforts by software personnel in developing reliable Protection class software, inappropriate use or maintenance of the software may undo the software reliability by the recipient after delivery. It is important that the recipient be trained and qualified to use or maintain the software. Software personnel shall be trained in the procedures in Section 11 involving exception reporting and correction.

3.5.1 Training

Common Q™ customers are responsible for providing safety training for the users, operators, and maintenance and management personnel, as appropriate. All training materials prepared for Common Q™ customers must be reviewed by the IV&V team per subsection 5.5.7

Westinghouse personnel assigned to work on any activity in the software life cycle process must complete training on the SPM in accordance with Reference 4.

3.5.2 Deployment

3.5.2.1 Installation

Installation documentation shall be developed, prior to the installation and checkout phase of the software life cycle, which will include the procedure(s) for installing the software. The IV&V team shall review this documentation according to the procedure in subsection 5.5.7.

3.5.2.2 Startup and Transition

Changes to installed systems may be disruptive to operations, particularly if problems occur or the resulting system operates differently. A Software Installation and Startup Procedure will be prepared addressing the following (as appropriate to the configuration of the system being installed):

- Fallback modes for the new system
- Startup of Backup components and subsystems
- Startup of the New system
- Parallel operation with backups
- Parallel operation of the old system and the new system
- Subsystem vs. full system operation
- Switchover to full system operation
- Validation of results from the new system
- Cross validation of results between the old system and the new system
- Fallback in the case of failure of the new system, including fallback to an old system if one exists

3.5.2.3 Operations Support

Documentation of the system and its software is supplied as described in Section 10. This documentation includes design documents, user manuals and instructions for maintenance expected by plant personnel.

3.5.3 Monitoring

Problem Reporting and Corrective Action (Section 11) contains requirements for monitoring the use of delivered software and associated exception reporting.

In addition, Protection class software is designed so that the integrity of the software can be verified periodically to detect unauthorized modification of code or data. Procedures necessary to perform this verification shall be documented. Methods shall be considered that provide automatic verification of the system during operation.

3.5.4 Maintenance

Software changes during all software life cycles are executed according to the Software Configuration Management Plan in Section 6 and the Software Maintenance Plan in Section 9.

3.5.5 Retirement and Notification

Subsection 6.2.2 describes the retirement of software and associated notification to current users.

(Last Page of Section 3)

SECTION 4
SOFTWARE QUALITY ASSURANCE PLAN

4.1 INTRODUCTION

4.1.1 Purpose

The Software Quality Assurance Plan (SQAP) describes the requirements and methodology to be followed in developing, acquiring, using, and maintaining software to be used for the design and operation of Common Q™ systems. The SQAP complies with Reference 13.

Software to be developed and used for the Common Q™ systems shall be placed into the following software classes (see subsection 1.2.1):

- Protection (safety critical)
- Important-to-Safety
- Important-to-Availability
- General Purpose

All software modules shall be developed or used consistent with the classifications shown in EXHIBIT 4-1 ASSIGNMENT OF COMMON Q™ SOFTWARE TO CLASSES for PPS/RPS, ESFAS, CPCS and PAMS. Common Q™ applications not listed in the exhibit shall document the software classification using the Safety Classification Record in accordance with the Requirements of Reference 4. Software that is initially assigned to one software class can be reassigned to another class provided that all tasks appropriate for the new class, up to the current phase of the software life cycle, are completed and satisfactorily reviewed. Changes in classification shall be documented via a Safety Classification Record in accordance with Reference 4. The Safety Classification Records are prepared by the design organization and are an input to the design and IV&V teams to determine the necessary requirements for design and IV&V activities. The appropriateness of the software safety classification is reviewed throughout the design and IV&V activities.

4.1.2 Scope

This SQAP is required for all quality classifications defined for the Common Q™ system: protection, important-to-safety, important-to-availability, and general purpose software.

This SQAP is based on the software life cycle model described in Reference 5 for Software Lifecycle.

Within each software class described in subsection 4.1.1, there are categories of software, which this SQAP addresses. These categories are described as follows:

1. Original software

2. Existing software

    a. To be modified
    b. Not to be modified

Documentation requirements depend on the classification and category of software and shall be consistent with EXHIBIT 4-1 ASSIGNMENT OF COMMON Q™ SOFTWARE TO CLASSES and EXHIBIT 4-3 TASKS REQUIRED FOR SOFTWARE CATEGORIES.

Existing software is software that has been created, but not under this SPM. To qualify for use under this SPM, the software must be evaluated by the design team to meet the following criteria:

- Existing commercial software may be used in protection and important-to-safety applications if it is qualified using a Commercial Grade Dedication Program (CGDP) such as the one described in Reference 3. To qualify existing commercial important-to-availability or general purpose software, the design team shall select applicable portions of the CGDP and qualify the software to those portions.

- Existing NPP non-commercial software that has been actively used in a nuclear power plant may be used for the same class of software under this SPM provided it has been maintained under an acceptable quality plan with an active program for problem and corrective action reporting. This software shall also have adequate design documentation, user documentation and well-commented source code. This software shall have been verified and validated under another program that is judged by the IV&V team to be acceptable.

- Other existing non-commercial software (i.e., source code freely available (e.g., freeware)) may be used under the following conditions:

    — This software can only be qualified as Important-to-Safety, Important-to-Availability, or General Purpose software.

    — The software fulfills a specific requirement identified in the Software Requirements Specification (SRS).

— The code is well organized and has adequate design documentation, and source code commentary. If the software has poor or no documentation then, documentation shall be prepared.

— Will undergo the IV&V process starting at the implementation phase.

For existing software that is qualified as above, design documentation and code may be used without revision to meet format or content requirements of this SPM. Modifications to this software may be made in accordance with prior documentation and code format.

Under this SQAP, a software product that is contracted for development by a subcontractor is treated as original software unless the software already exists and is in use. In this case, it is treated as existing software.

This SQAP describes the methodology by which all software and associated documentation is managed throughout the life cycle. Software elements produced in the process of quality assurance are as follows:

- Test plans, cases, procedures and reports
- Review and audit results
- Exception reports and corrective action documentation
- Software configuration management plans
- Software verification and validation plans

4.1.3 Software Development Process

The software development process for original software is shown in EXHIBIT 4-2 COMMON Q™ SOFTWARE DEVELOPMENT PROCESS. This exhibit shows the relationship between software and hardware, the process of software integration and testing, the design documentation produced, and the quality assurance documentation required throughout the software life cycle.

As shown in EXHIBIT 4-2 COMMON Q™ SOFTWARE DEVELOPMENT PROCESS, software quality is assured through the process of verification reviews, validation testing at the different stages of development, and software configuration management during all phases of software development. Software Verification and Validation activities are governed by the Software IV&V Plan described in Section 5. Required test procedures and test reports are shown in the exhibit, and are based on the level of the test and the class of the software.

4.2 REFERENCES

Refer to page xx for a list of references.

4.3 MANAGEMENT

The management of all software for Common Q™ projects spans the software life cycle defined in subsection 1.4.1 and applies to all software classes described in subsection 4.1.1.

4.3.1 Organization

The implementation of an effective SQAP is the responsibility of all persons involved in the software development process. Each person responsible for the software development shall perform their work in accordance with established standards, methods, and procedures identified in this SQAP.

Software life cycle activities for a Common Q™ project shall be performed by the Nuclear Automation Organization (NA) described in Section 2. A design team, an IV&V team, and a Quality organization are responsible for the execution of all quality assurance tasks.

The NA organization includes a Quality organization and an Engineering organization. The design team and the IV&V team are in separate organizations at least to the Director Level. The design team is responsible for the software design and implementation, software quality assurance planning, and software configuration management. The IV&V team is responsible for software design verification, and software validation testing. The two teams are independent from one another as depicted in EXHIBIT 2-1 DESIGN/IV&V TEAM ORGANIZATION.

The Quality organization is responsible for coordinating and reviewing quality assurance procedures and directives. The Quality organization has a reporting chain separate from the design team such that the QA organization is independent of project schedule and cost considerations. The Quality organization provides oversight by way of periodic audits to verify that the NA organization is correctly abiding by both the procedures and directives generated by both organizations.

The Engineering Project Manager (EPM) shall be responsible for all design team activities being in accordance with this SQAP. Verification of the implementation of quality assurance requirements is performed by the Quality organization in accordance with References 1 and 4.

The IV&V Team Leader shall verify that software and associated documentation has been developed in accordance with the standards specified in this SQAP. This includes ensuring that the coding standards (subsection 4.5.2.1), testing standards established in the test plan and documentation standards (Section 10) have been followed.

In general, software configuration management responsibilities span all phases of the software life cycle for

- Development of Software Configuration Management Plans
- Execution of software configuration management activities per the SCMP
- Control of software through a librarian
- Baselining and integration of new software versions

4.3.2 Tasks and Responsibilities

This section describes the specific tasks and responsibilities to be performed by the Nuclear Automation design and IV&V teams. All tasks and responsibilities described in this section apply to each Common Q™ project. Tasks are listed in the life cycle phase for which they will be performed. Typical tasks are: software design and development, software quality assurance planning, verification reviews, audits, test planning, test execution, and test reporting. Tasks required are based on software category. EXHIBIT 4-3 TASKS REQUIRED FOR SOFTWARE CATEGORIES shows the software tasks for each category in each phase.

The following are some procedural types of actions that are performed to confirm traceability throughout the development and verification stages:

1. The software design documents are dated and signed by the designer and the design team leader.

2. Each software release record is dated and signed by the programmer or design team leader.

3. The corresponding Common Q™ software verification report and software test procedures documents are dated and signed by the IV&V author and the IV&V team leader.

4. Each protection class software module test report is verified, dated, and signed by the tester.

5. A configuration status accounting of software is maintained to effectively manage the software configuration.

4.3.2.1 Initiation (Concept) Phase

Common Q™ system software quality assurance planning shall be performed during this phase. A Project Quality Plan (PQP) (Reference 4) shall be developed. Any alternatives to the SPM processes or additional project specific information for the SQAP, SVVP, SCMP or SMP shall be documented and justified in the PQP. The PQP author shall also define, or reference the applicable coding standards

within the PQP. The IV&V team reviews the design team's outputs during this phase. Any anomalies found will be documented using Exception Reports.

4.3.2.2 Software Requirements Phase

The Common Q™ system Software Requirements Specification (SRS) is developed during this phase. Input from the system requirements specification provides the necessary system and functional requirements to develop software requirements and hardware design. The system requirements specification is used to generate equipment specifications and software documents. These system requirements are noted in EXHIBIT 4-3 TASKS REQUIRED FOR SOFTWARE CATEGORIES.

The design team shall be responsible for developing, maintaining, and updating its SRS. A separate SRS shall be developed for each Common Q™ system based on system requirements, and shall provide the detail and information sufficient to design the software. The SRS shall be divided to describe software requirements for the software in each class in the system. The SRS shall be developed in accordance with subsection 10.2.2 of this SPM.

The IV&V team, as shown in EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES, shall verify each SRS. The verification review shall confirm that the system requirements are properly reflected in the SRS. Verification of SRSs shall be performed in accordance with subsection 4.6.2.1.

A Common Q™ specific test plan shall start to be developed in accordance with the content, not the format of Reference 14, Section 4 to identify how the test activities will be implemented. It shall include the following topics as a minimum:

- General approach including: identification of test procedures, general test methods, documentation of results, and traceability methods to the SRS and SDD.
- Requirements for testing including: test boundary conditions on inputs and unexpected input conditions.
- Test management including: personnel, resources, organization, and responsibilities.
- Procedures for qualification and control of the hardware to be used in testing.
- Qualification and use of software tools.
- Installation test requirements for existing software that is used without modification.
- Regression test requirements for previously qualified software to be modified.

- Delineate major features of the system that will be tested.

The IV&V team reviews the design team's outputs during this phase. Any anomalies found will be documented using Exception Reports.

4.3.2.3 Software Design Phase

The design team shall be responsible for developing, maintaining and updating a Software Design Description (SDD) for each software module. Each SDD shall be traceable to the requirements set forth in the SRS, and shall include enough detail to begin coding in the Implementation Phase. All SDDs shall be developed in accordance with the requirements of Section 10.3.

The IV&V team as indicated in EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES shall verify each SDD. The verification review shall confirm that the software requirements identified in the SRS are properly reflected in the SDD and that the SDD is reflected in the RTM. Verification of SDDs shall be performed in accordance with subsection 4.6.2.2.

Prototype software may be developed to prove a new principle or to help further define the software design during this phase. Prototype software has a different software life cycle than the other categories of software that is usually shorter in duration. Specifically, prototype quality assurance tasks shall include:

- Adherence to coding standards
- Documentation of prototype design (format at the discretion of the design team)
- Informal verification reviews
- Limited software configuration management

Wherever prototype software is reused and integrated into the deliverable software, it shall undergo the respective software quality measures based on its software class. This includes software quality assurance tasks described above from the integration point forward in the life cycle plus any "skipped" tasks in the life cycle for; verification reviews, audits, software configuration management activities, required documentation, and conformance to coding standards.

The IV&V team reviews the design team's outputs during this phase. Any anomalies found will be documented using Exception Reports.

4.3.2.4 Software Implementation Phase

Original software development and modifications to existing software shall begin with module coding by the design team in accordance with the appropriate coding standards listed in subsection 4.5.2.1.

Existing software, which has been qualified as described in subsection 4.1.2, may be integrated into the software system and tested during this phase.

Verification of module code shall be performed by the group identified in EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES. Details of software module code verification are described in subsection 4.6.2.3.

Validation of software during this phase can be accomplished by several methods. Some possible methods are identified below:

- One method is to hierarchically assemble the modules into units and perform a unit test, and subsequently assemble all the units into the system and perform integration and system validation testing. Protection class software requires formal module testing.

- Or, the test sequence can be performed in a series of expansions. This could be accomplished by continually adding successfully tested modules to the "system" and test after each addition until the complete system is assembled and tested.

Validation of software at module and unit level shall be performed in accordance with Section 7, which is in compliance with Reference 12. Internal state testing is conducted during module testing. The responsibility for testing will be assigned to the design team or IV&V team, as shown in EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES. Unit test procedures and reports are only required for software classified as protection and as important-to-safety. Module test procedures and reports are only required for software classified as protection.

The IV&V team reviews the design team's outputs during this phase. Any anomalies found will be documented using Exception Reports.

4.3.2.5 Testing Phase

System validation testing shall be conducted during this phase in the development environment when all of the system components (and system boundaries) have been integrated by the design team per the project Test Plan. The purpose of this test is to evaluate the system as a whole for its ability to meet system usage and performance requirements. Test procedures and reports shall be documented in accordance with Section 5.8, and verified by the groups identified in EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES. The groups identified in the exhibit shall conduct system tests.

The Final Software Verification and Validation Report (SVVR) for the deliverable software shall be prepared during this phase. All user Documentation shall be developed during this phase in accordance with Section 10. Also, during this phase, software load instructions shall be verified by the IV&V team.

The IV&V team reviews the design team's outputs during this phase. Any anomalies found will be documented using Exception Reports.

4.3.2.6 Site Installation and Checkout Phase

Site installation and Checkout of Common Q™ software will be dependent on the contractual arrangements made with the customer that purchased the specific Common Q™ system. If Westinghouse is responsible for software installation and checkout then the design team shall have the responsibility for the Site Installation and Checkout phase and the IV&V team shall be responsible for associated IV&V requirements.

The preparation of the site test plan will be initiated during or after the requirements phase to support evaluation of requirement testability on-site. Validation of the installed software shall be performed to determine that the software was installed correctly. Software installation validation applies to initial software and any subsequent revisions.

During this phase the software becomes part of the installed equipment incorporating applicable software components, hardware, and data. The process of integrating the software with applicable components in the plant consists of installing hardware, installing the software, and verifying that all components have been included.

If within Westinghouse's scope of supply, an Exception Report Log shall be maintained during the installation and checkout phase in accordance with the Site Acceptance Test (SAT) plan. This log shall be verified by the IV&V team after installation for Protection and Important to safety class software.

After installation, the equipment and software shall be checked out, according to the SAT plan and procedure. All test exceptions shall be documented using the Exception Report form and entered into the Exception Report Log.

In this phase, the site portion Software Verification and Validation Report (SVVR) shall be prepared for protection class and important-to-safety class software. Details of the SVVR are described in Section 10.5.

4.3.2.7 Operations and Maintenance Phase

Activity in this phase consists of maintenance of the software to:

- Remove identified latent errors
- Respond to new requirements, or
- Adapt the software to changes in the operating environment.

Software modifications shall be approved, documented, verified and validated, and controlled in the same manner as described previously in the Design, Implementation and Test Phases. The SVVP (Section 5), in conjunction with the SCMP (Section 6), shall also be used to assist in the management of these activities and procedures.

4.4 DOCUMENTATION

4.4.1 Purpose

The documentation required for each category of software is listed in EXHIBIT 4-3 TASKS REQUIRED FOR SOFTWARE CATEGORIES. Section 10 of this SPM provides guidance for the development of documents. If required, documents listed shall be made lifetime quality records in accordance with Reference 4.

4.5 STANDARDS, PRACTICES, CONVENTIONS, AND METRICS

4.5.1 Purpose

The standards, practices and conventions to be applied to the Common Q™ systems are contained in Reference 1. Compliance with these standards shall be monitored and assured through the review and audit process described in Section 4.6. Additional detailed instructions that may be required to implement the software development process should be implemented as Work Instructions in accordance with the requirements in Reference 4.

4.5.2 Content

4.5.2.1 Coding Standards

The software development process shall provide guidance to promote standardization, compatibility and maintainability of resulting software products. The process shall provide a coding standard for each language, database, or software tool that allows author discretion in establishment or use of convention. The coding standard shall also include the commentary and logic structure standards. Coding standards to be applied to a project shall be referenced in the Project Quality Plan. The IV&V team shall review the applicable coding standards for each project for acceptability. The IV&V team shall assure that the Common Q™ project uses IV&V approved coding standards. If IV&V is a signatory on the generic Common Q™ coding standards, then this represents an evaluation of the acceptability of these standards for all Common Q™ projects.

This requirement applies to the following typical software products:

- Assembly languages
- C/C++
- Display building languages
- Function Block Diagrams

Each coding standard shall contain, but is not limited to, the following information:

1. General

    - This area outlines general ideas and concepts used to guide the creation of software written under a specific language.

2. Naming conventions

    - Filename extensions as far as how they are used to organize files.
    - Information pertaining to file organization within a system.
    - Variables naming

3. Internal documentation guidelines

    - Program identification header content, placement, type, quality, and quantity.
    - Revision history recording within each source file.

4. Stylistic conventions

    - Issues that affect readability, such as indentation and use of white space.

5. Use of specific language features

    - List forbidden or restricted functions

6. Software tool usage guidelines

    - Information and use of automatic make facilities
    - Appropriate compiler flag usage

7. Functions

    - Modularity
    - Naming

4.5.2.2 Software Testing Standards

Software testing methodologies, policies and practices shall be described in the project specific Test Plan. Specific format and content for test procedures and test reports shall also be provided in the Test Plan and shall comply with Section 5.8.

4.5.2.3 Documentation Standards

All documents developed for Common Q™ systems shall comply with the requirements for format and content described in Section 10.

4.5.2.4 Metrics

The following metrics shall be maintained for each Common Q™ system:

1. The errors discovered during Integration testing shall be identified using the information required by EXHIBIT 11-1 EXCEPTION REPORT so the number of errors discovered can be tracked for error discovery metric reporting. The overall goal is to identify a decreasing number and severity of errors as the testing progresses from Integration testing to system validation and FAT to SAT. The exhibit represents the minimum information required. The exception reporting procedure shall be implemented via an automated process.

2. System validation and FAT errors shall be reported through the use of Exception Reports and the number and severity shall be identified for error discovery metric reporting.

3. Software errors discovered after FAT and before SAT shall be tracked through the use of Exception Reports, and the number and severity shall be identified for error discovery metric reporting.

4. Software errors discovered during SAT shall be tracked through the use of Exception Reports and the number and severity shall be identified for error discovery metric reporting.

5. Software errors discovered after SAT (after system acceptance) shall be tracked and the number and severity shall be identified for error discovery metric reporting.

4.6 REVIEWS

4.6.1 Purpose

The purpose of this section is to address the review requirements throughout the software life cycle.

The software reviews required by this SQAP address software classes and categories described in subsections 4.1.1 and 4.1.2.

Reviews are technical in nature and are designed to verify the technical adequacy and completeness of the design and development of the software.

Review activities applicable for each Common Q™ project include the following:

- Software Requirements Review (SRR)
- Software Design Review:

  — Architecture Design Review
  — Critical Design Review

- Code Verification
- Software Verification and Validation Plan (SVVP) Review
- Functional Review
- Physical Review
- In-process Audits
- Managerial Reviews
- Software Configuration Management Plan (SCMP) Review
- Post Mortem Review

The reviews, the group responsible for the reviews and the methodology for performing the reviews are defined herein. Peers who have an equivalent knowledge of the topic but who are not directly involved with the application as required in Section 2 shall perform the reviews.

Audits are designed to confirm that software documentation and processes comply with the established standards and guidelines set forth on the project.

References to the SVVP are provided in this section to address specific areas of the review and audit process. In some cases, the procedural aspects of the review are contained in the SVVP. The reviews defined in IEEE 1028 (Reference 16) are either conducted by the IV&V team per the requirements of this SPM, or by QA or Management in accordance with Westinghouse Level 2 procedures (Reference 4) of the NRC-accepted Westinghouse Quality Management System. The following reviews called out in Reference 16 are conducted as follows:

1. Management Reviews – Monitoring progress and determining the status of plans and schedule are performed in accordance with Reference 4, W2-10.2-101 of the NRC-accepted Westinghouse

Quality Management System. Confirming requirements and their system allocation is performed in accordance with subsection 5.5.3.

2. Technical Reviews – Performed in accordance with Reference 4, W2-8.4-101 and W2-8.4-102 of the NRC-accepted Westinghouse Quality Management System, and the IV&V requirements in this SPM.

3. Inspections – Performed by IV&V in accordance with Section 5.

4. Walk-throughs – Performed in accordance with Reference 4, W2-8.4-101 of the NRC-accepted Westinghouse Quality Management System, and the IV&V requirements in this SPM.

5. Audits – Performed in accordance with Reference 4, W2-4.2-101 of the NRC-accepted Westinghouse Quality Management System, and to some extent the IV&V requirements in this SPM.

4.6.2 Minimum Requirements

Reviews shall evaluate specific software elements (such as files, functions, modules, or complete systems) to confirm that the requirements are adequate, technically feasible and complete. The following subsections define the minimum review requirements.

4.6.2.1 Software Requirements Review (SRR)

After the design team has completed the requirements phase, the IV&V team shall conduct the SRR. It shall examine the Software Requirements Specification (SRS) to verify that it is clear, verifiable, consistent, modifiable, traceable, and usable during the operations and maintenance phases. The SRR shall include an evaluation of the software requirements against the user's software application, which is described in a higher level requirements document such as a system requirements specification.

Specific SRR items are described in Section 5 and shall be described in detail as necessary in the SVVP. As a minimum, these items shall include:

- Traceability and completeness of the requirements
- Adequacy of rationale for derived requirements
- Testability of functional requirements
- Adequacy and completeness of verification and acceptance requirements
- Conformance to documentation standards
- Adequacy and feasibility of performance requirements
- Adequacy and completeness of interface requirements

Responsibilities, methodologies, and reporting of results are described in Section 5 and shall be described in detail as necessary in the SVVP. Frequently encountered categories or types of errors normally found in the SRS may also be included in the SVVP in order to aid the independent reviewer.

4.6.2.2 Software Design Review

4.6.2.2.1 Architecture Design Review

After the initial issuance of the SDDs, the IV&V team shall conduct the Architecture Design Review (ADR) of the software. It shall include a review of the preliminary SDD and RTM, emphasizing the following issues:

- Detailed functional interfaces with other software, system equipment, communication systems, etc.
- Software design as a whole emphasizing allocation of software components to function, functional flows, storage requirements and allocations, software operating sequences, and design of the database
- An analysis of the design for compatibility with critical system timing requirements, estimated running times and other performance issues
- Human factor requirements and the human machine interfaces for adequacy and consistency of design
- Testability of design
- Technical accuracy of all available test documentation and its compatibility with the test requirements of the SRS
- General description of the size and operating characteristics of all support software
- Description of requirements for the operation of the software
- Identification of requirements for functional simulation, environmental recording, configuration, etc.

The results of the review shall be documented in the IV&V report, identifying all deficiencies found during the review. The design team shall plan and schedule any corrective actions required.

4.6.2.2.2 Critical Design Review

After the design team has completed the design phase of the project, the IV&V team shall conduct the Critical Design Review (CDR). It evaluates acceptability of the detailed design documented in the SDD, and establishes that the detailed design satisfies the requirements of the SRS. The review also verifies the design's compatibility with the other software and hardware that the product is required to interact with and assesses the technical risks of the product design.

The CDR shall include a review of the SDD and available test documentation for the following items:

- The compatibility of the detailed design with the SRS
- Available data in the form of logic diagrams, algorithms storage allocation charts, and detailed design representations
- Compatibility and completeness of interface requirements
- All external and internal interfaces including interactions with the database
- Technical accuracy of all available test documentation and its compatibility with the test requirements of the SRS
- Requirements for the support and test software and hardware to be used in the development of the product
- Final design including function flow, timing, sizing, storage requirements, memory maps, database, other performance factors

The results of the review shall be documented using the IV&V Design Phase Checklist and should describe all deficiencies identified in the review. The design team shall plan and schedule any corrective actions required. After the SDD is updated to correct any deficiencies, it shall be placed under configuration control to establish the baseline to be used for the software coding.

4.6.2.3 Code Verification

Software code shall undergo periodic peer review by means of a code inspection. Code reviews are performed by an independent reviewer from either the design team or the IV&V team. Code reviews shall verify that the source code conforms to the software coding standards and guidelines described in subsection 4.5.2.1. Code reviews shall include evaluation of the source code implementation against the SDD. The review criteria are specified in EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION DESIGN PHASE CHECKLIST.

4.6.2.4 Software Verification and Validation Plan Review

The SVVP (Section 5) is reviewed for adequacy and completeness of the verification and validation methods defined in the SVVP. An independent reviewer meeting the qualifications of Reference 4 performed this review as part of the review process for this SPM. Compliance to the SVVP is covered by the in-process audits described in subsection 4.6.2.7.

4.6.2.5 Functional Review

After the test phase, the IV&V team shall conduct the Functional Review. It is conducted prior to software delivery to verify that all requirements specified in the Software Requirements Specification have been met. The review shall include an overview of all documentation and a review of the results of

previous reviews, including Software Requirements Review, ADR, CDR, and if applicable, interim IV&V reports (for Protection and Important-to-Safety class software).

Any findings in the Functional Review shall be documented in the final IV&V report.

4.6.2.6 Physical Review

Physical Reviews are held to verify that the software and its documentation are internally consistent and are ready for delivery. It is when the IV&V Final Report is issued that the software and documentation are considered internally consistent and ready for delivery.

The IV&V team produces the deliverable software media and the EPM confirms that the deliverable software media is in conformance with customer requirements.

The IV&V team shall also verify that the software change control process was adequately followed.

4.6.2.7 In-Process Audits

In-process audits of a sample of the design are held to verify consistency of the design process. The Quality organization shall perform in-process audits for Common Q™ systems for software classes Protection and Important-to-Safety. The audit shall review different items depending upon the software phase in progress when the audit is held and can include a review of the following items:

- Compliance with this Software Program Manual including the documented evaluation of the following required activities performed by the design and IV&V team:
  - Code versus design documentation (code walkthroughs or code inspections)
  - Interface specifications
  - Design implementations versus functional requirements
  - Functional requirements versus test description
  - Test descriptions versus test procedures
  - Test procedures versus test reports

The results of in-process audits shall be documented identifying all deficiencies found. The EPM, or designee, shall evaluate the deficiencies, identify corrective actions, and define schedules for resolving the deficiencies.

4.6.2.8 Managerial Reviews

As part of the Quality organization responsibility, it shall either perform or facilitate this review. The purpose of this review is to assess the execution of all of the actions and the items identified in this SQAP.

The managerial review shall be documented by a report summarizing the review findings, exceptions to the process stated in the SQAP and recommended changes or improvements to the SQA process. The reviews result in statement as to the adequacy of the SQA process and its execution.

4.6.2.9 Software Configuration Management Plan Review

The Design Team shall identify adherence to the Software Configuration Management Plan in this SPM and make note of any augmentations or deviations in the project plan.

The Software Configuration Management Plan (SCMP) Review is held to evaluate the adequacy and completeness of the configuration management methods defined in the SCMP (Section 6) and their implementation. By IV&V signoff of this SPM, the SCMP (Section 6) was reviewed and found acceptable by IV&V. Any comments resulting from their review have been incorporated. The IV&V team shall review and document the design team's adherence to the SCMP for each Common Q™ project.

4.6.2.10 Post Mortem Review

The EPM shall conduct a project closeout review upon completion of the project to confirm that all project activities have been completed, all deliverables have been shipped, and that all project quality assurance activities have been fulfilled. Project metrics should be reviewed at this time to determine if any process improvements can be identified. Suggestions for improvement and/or best practices that are identified during the Post Mortem Review should be documented in accordance with Reference 4, Westinghouse Corrective Action Program of the NRC-accepted Westinghouse Quality Management System. Customer satisfaction surveys may also be initiated.

4.7 TEST

Required testing to be performed for all software related projects includes:

- Module level tests (Documented module tests are required only for protection class software.)
- Unit level tests (Documented unit tests are required only for protection and important-to-safety class software.) (Can be part of Integration and System Validation Tests)
- Integration Tests
- System Validation Tests*

- Factory Acceptance Tests (FAT)**
- Site Acceptance Tests (SAT)**

* The System Validation Test encompasses the scope of FAT, so there is no need to conduct FAT as a separate test on a first-of-a-kind system.

** Subsequent systems of the same design would only undergo these tests.

4.8 PROBLEM REPORTING AND CORRECTIVE ACTION

4.8.1 Purpose and Scope

The purpose of a formal procedure of software exception reporting and corrective action is to confirm that all software errors and failures are promptly acted upon and in a uniform manner encompassing all project software. This procedure ties together the requirements of the SVVP and the SCMP. IV&V activities are the primary vehicle to uncover software problems, while the SCMP shall describe actions taken to correct problems by changing configured software are consistent and traceable.

Exception reporting and corrective action procedures shall span the entire software life cycle and all software classes identified in this SQAP. These procedures are detailed in Section 11 of this SPM.

4.9 TOOLS, TECHNIQUES AND METHODOLOGIES

Software development for Common Q™ projects shall use a number of techniques to help assure all software is designed, implemented, and documented in accordance with the Common Q™ objectives of building software which meets the requirements and which is maintainable over time in the most cost effective manner. The tools, techniques and methodologies employed in this process shall provide the means for the software to be verifiable from each phase of the project to the next.

- Use of structured design techniques for analyzing and developing the software design. These shall include data flow diagrams, where applicable, to represent the interactions among modular elements and the flow of data among them. Entity-relation charts may be used to represent any relational database structures.

- NA management sign-off and approval of all design and IV&V documentation shall include one of the following:

  — The ELM of the author, or
  — The EPM

- All members of the Common Q™ design and IV&V teams shall be trained in the contents of this SPM. This training shall be documented in the individuals' training records.

- Use of the waterfall model of software development and testing techniques to help assure that the requirements are correctly translated into design and implementation products.

- The use of commercially available automated tools for software configuration management should be employed to the maximum extent possible.

4.10 CODE CONTROL

Code Control shall be provided as part of software configuration management per Section 6. Methods and facilities used for maintenance, storage, documentation and security for controlled versions of the software during all phases of the software life cycle are also defined in Section 6.

All software items shall be controlled to maintain the items in a known and consistent state at all times. New software and modifications to existing software shall follow the configuration requirements for all life cycle phases. Existing software, which is not to be modified, including tools used in the software development, test, and documentation process, shall be placed under configuration control procedures upon its introduction or use within the software system.

4.11 MEDIA CONTROL

The methods and facilities used to protect computer program physical media from unauthorized access or inadvertent damage or degradation are described herein.

4.11.1 Media Identification

Media identification is described in subsection 6.3.1. Removable storage media should not be switched, renamed, or initialized without prior approval from the EPM, or designee.

4.11.2 Archival Requirements

A locked storage facility shall be used to store all project software deliverable physical media in a location separate from the configuration management (version control) server the deliverable was created from or digitally stored. This locked storage facility shall be able to accommodate the storage of all utilized types of physical media.

After important Common Q™ software development milestones or baseline configurations are archived in a version control system, a known software configuration shall be completely backed up and periodically stored in a data storage area separate from the software development area.

The requirements in this section can be performed by the software librarian. The software librarian may initiate the setup and maintenance of periodic digital backup of the safety system software configuration through requests to the Information Technology department.

4.12 SUPPLIER CONTROL

The purpose of this section is to describe the level of software quality assurance measures to be applied to software supplied to a Common Q™ system from parties outside of Westinghouse.

4.12.1 Existing Software

This SQAP defines existing software as software which was previously developed prior to the Common Q™ system being developed, to satisfy a general market need and may be considered for use on a Common Q™ project. The software may be subsequently modified prior to delivery, or it may be used "as is."

Existing software includes commercial software that is integral to the delivered system and software that is determined to be in support of the delivered system. Examples of integral software would be:

- Operating systems
- Compilers, Linkers, Loaders
- Database software
- Communication Drivers
- Man-Machine Interface software
- Display building software

All commercial software that will be used for Protection and Important-to-Safety class software in Common Q™ protection systems must meet the requirements established in a Commercial Grade Dedication Program like the one described in Reference 3.

For existing software, which is modified for a Common Q™ project, all software requirements specified in this SQAP for original software shall be in effect for the modifications. The minimum IV&V activities applicable for modifications to existing software are: software modification requirements verification, software modification design verification, program modification documentation verification, and software validation. Regression testing using test cases shall be conducted to validate that the modifications do not

produce unintended adverse effects, and to validate that the modified software still meets the original software requirements.

Existing software that is not modified shall be qualified for use according to subsection 4.1.2.

Once qualified for use, the software shall fall under the Common Q™ SCMP (Section 6). Once installed, the software shall meet the following requirements:

- Verification and Validation during Installation and Operation per the SVVP,
- Configuration Management during Installation and Operation per the SCMP,
- Documentation including: Test Plans, Procedures, SVVR, and User Manuals,
- Exception reporting and corrective action procedures, and
- Records of delivered documents and software

4.12.2 Sub-Contracted Software/Services

Original software for Common Q™, that is developed by a contractor and purchased, shall adhere to the quality assurance requirements specified in this SQAP for original software. This applies regardless of whether the software will be subsequently modified or not. This does not apply to software in systems that are commercially dedicated.

Additional requirements for subcontracted software and services are as follows:

- Software and services must be procured from approved supplier, per Reference 4.

- Suppliers must have written quality assurance policies that meet the principles and intent of this SQAP.

- Purchase orders shall require the Supplier to make available documents that provide evidence of compliance with the principles and intent of this SQAP.

- Purchase orders shall require the Supplier to deliver adequate user documentation, test procedures and test reports.

- In-house contractors will follow all internal training procedures.

- An external monitoring program shall be in place to confirm that subcontractors adhere to the requirements of this SPM.

4.13 RECORDS COLLECTION, MAINTENANCE AND RETENTION

Records collection, retention, and maintenance shall be in accordance with Reference 4.

4.14 TRAINING

All design and IV&V team members involved with Common Q™ software shall be trained on the Software Program Manual (either by classroom training or self-study). The individual's training record shall be used as documentation that this training took place.

4.15 RISK MANAGEMENT

Reference 4 describes the process and requirements for risk management for project execution.

(Last Page of Section 4)

SECTION 5
SOFTWARE VERIFICATION AND VALIDATION PLAN

5.1 PURPOSE

The purpose of this section is to establish requirements for the IV&V process to be applied to Common Q™ systems. It also defines when, how and by whom specific IV&V activities are to be performed including options and alternatives, as required. The section includes various IV&V methodologies aimed to increase the system reliability and availability. Some of these methodologies employ systematic checks for detecting errors in the software and hardware interface, during the system development and implementation process. This section explains requirements for the IV&V processes starting with the system design document stage and all necessary IV&V activities to verify and/or validate I&C systems. This SVVP complies with Reference 8 requirements for V&V activities. A table that shows how this SPM meets the requirements of Reference 8 is included in EXHIBIT 5-8 IEEE STANDARD 1012-2004 COMPLIANCE TABLE.

The goals of this IV&V plan, when applied to a specific project, are to:

- Improve the system reliability and availability
- Reduce system costs by exposing errors as early as possible
- Provide a systematic process of objectively evaluating the system's performance
- Demonstrate compliance with customer requirements, industry standards and licensing requirements

5.1.1 Categorization of Software Items and Review Scope

IV&V is performed on documents and materials that are produced according to the category of each software item, as described in Section 4. For example, a software design description is not required for an existing commercial off-the-shelf software package. IV&V activities only include documents and materials identified in Section 4.

5.1.2 IV&V Program Implementation

IV&V activities are integrated into the requirements, design, implementation, test and installation phases described in Section 4. Experience has shown that the earlier a deficiency is discovered, the easier and more economical it is to resolve. The initial activity is the review of system functional requirements prior to any detailed software design. Verification activities are performed at the end of this phase, and each

subsequent phase. These activities determine that all requirements have been properly transferred from the input products to the output products of the phase, with amplifications or modifications appropriate to the phase. Upon completion of the software implementation, validation activities are performed. These activities determine that the operation of the system is consistent with the system requirements. Thus IV&V activities are integrated with project activities from the beginning to end.

Once a system design and implementation has been verified and validated, any succeeding systems manufactured of the same design are certified by standard manufacturing test procedures. Some of the tests used by manufacturing are the same or equivalent to those used in the original system IV&V process. The manufacturing test is comprised of hardware functional tests and a Factory Acceptance Test (FAT). FAT is a subset of the Integration and System Validation Testing. System Validation Testing is not repeated on these succeeding systems manufactured of the same design. If System Validation Testing is conducted on a delivered system, a separate FAT does not need to be conducted given that the System Validation tests fully exercises the hardware as well as the software being delivered. The documentation for the tests performed on manufactured units is maintained under configuration management control. Any design changes that would impact manufactured units are re-verified and maintained under configuration management control.

5.1.3 Prominence of IV&V Documentation

Traceability is important, not only to document the IV&V activities, but also to record appropriate actions taken to resolve discrepancies. Thus an IV&V program is, by its nature, oriented heavily towards documentation and the ability to trace changes in project documents. All comments generated by the IV&V team and all comment resolutions shall be documented consistent with EXHIBIT 11-1 EXCEPTION REPORT. Section 10 defines the structure and format of the documents that may be produced during various phases of the project. The documents' contents will vary depending on the specifics of system or project; however a system to trace the documentation and deficiency resolution is required. In the early phases of the system design process the system is divided into manageable modules of software and hardware. In the later phases, these modules are integrated into a total system.

The Configuration Management Plan addresses these issues and details (1) how the documents are controlled, (2) how records of changes and distribution are maintained, and (3) status of each document is identified.

5.1.4 Overall Common Q™ and Project-Specific IV&V Plans

This Common Q™ IV&V plan details the IV&V process and activities involved during the various phases, and details various tools and techniques to be used. Any deviations or additional project specifics to the SVVP, such as scheduling specific IV&V tasks and resource identification, shall be

defined in either a Project Plan or in a project-specific IV&V plan that is referenced by a Project Plan, as described in Reference 4.

5.2 REFERENCED DOCUMENTS

Refer to page xx for a list of references.

5.3 DEFINITIONS

Refer to page xiii for a list of acronyms and trademarks. Refer to page xvi for definitions.

5.4 VERIFICATION AND VALIDATION OVERVIEW

5.4.1 Organization

An independent IV&V team performs the safety activities for a given Common Q™ system implementation project. The IV&V team performs software design verification, software validation testing and software configuration status accounting activities on the Common Q™ systems.

The degree of independence required by this plan varies with the software classification. The applicability of the tasks varies with the software category. The general definition of and qualifications for reviewer independence are stated in Reference 4.

The IV&V team is organized independently of the design team. IV&V team members may not participate in any design team activities, but may participate in walk-through activities described in subsection 4.6.1. Also, the IV&V team leader, responsible for the IV&V, shall be organizationally independent from the design team leader. EXHIBIT 2-1 DESIGN/IV&V TEAM ORGANIZATION shows the relationship between the design team and the IV&V team. The IV&V team reports to an Engineering Line Manager (ELM), who is administratively and financially independent from the design team manager.

The reviewers of software in non-safety critical classes may be members of the requirements team or, in some cases, the design team. Nevertheless, the review of any particular software item shall not be performed by the individual(s) responsible for the requirements or design of the item. An independent reviewer must also be one who can perform a competent review.

EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES identifies the minimum review independence required for each type of document or software item, for each class of system.

5.4.2 Master Schedule

The Project Quality Plan (described in Reference 4) shall include the project IV&V schedule and required milestone delivery dates. This shall be developed in coordination with the IV&V team leader (for a IV&V team of more than one person).

5.4.3 Resources Summary

5.4.3.1 Design Team

Design team members organizationally report to an Engineering Line Manager (ELM). The ELM provides resource management of people and other resources (such as materials and equipment) to provide optimal implementation of customer projects for their assigned products and services. The composition of the design team shall be established in terms of the functions that are required within the team. One or more people depending on project size and complexity fulfill the following functions.

5.4.3.1.1 Lead Engineer

This is the team leader, responsible for all technical matters in the development of the system. Normally one person is designated as the lead engineer for a project. The lead engineer shall have the responsibility for the development of the software design requirements and software design specification documents. Global decisions on the structure of the software, decomposition, and database are made by the lead engineer. Some critical sections of the programs, both in terms of importance and complexity, may be coded by the lead engineer. The lead engineer supervises the rest of the design team in technical matters.

5.4.3.1.2 Programmer

A programmer's main responsibility is to develop the code and provide the details for the software design at the module level to meet the software design requirements. In most projects, it is anticipated that there will be more than one programmer.

5.4.3.1.3 Language Expert

This team member supplies the technical information on the programming language that is used. This person is preferably one of the programmers.

5.4.3.1.4 Hardware Expert

The hardware expert's responsibility is to maintain all hardware in working order in the "as delivered" system configuration. The hardware expert should also have software experience in order to assist in writing software drivers. There could be more than one hardware expert per project.

5.4.3.1.5 Engineering Project Manager

The Engineering Project Manager (EPM) is assigned to a particular Common Q™ customer project and is responsible for the development, scheduling, and the financial and quality execution of the assigned project. The Common Q™ Platform Lead may be responsible for these functions for internal generic Common Q™ development activities. The Common Q™ Platform Lead is responsible for the platform development meeting the continuing needs of the product family. Organizationally, EPMs and Platform Leads directly report to an Engineering Line Manager (ELM). EPMs and Platform Leads may delegate the performance of necessary tasks to other persons but remain responsible for their execution.

5.4.3.2 Independent Verification and Validation Team

IV&V team members organizationally report to an Engineering Line Manager (ELM) who is administratively and financially independent from the design team manager. The IV&V team ELM provides resource management of people and other resources (such as materials and equipment) to provide independent implementation of IV&V tasks. The composition of the IV&V team shall be established by the functions carried out, similar to the manner of the design team. The following functions are fulfilled by one or more people depending on project scope and complexity.

5.4.3.2.1 IV&V Team Leader

The IV&V team leader is responsible for all technical and administrative matters concerning the verification of the system. The IV&V team leader is responsible for the development of the verification requirements and validation test procedure documents. It is also the responsibility of the IV&V team leader to check the documentation compiled by the design team to the requirements.

5.4.3.2.2 Verifiers

The Verifiers check the portions assigned to them with the use of the project validation test procedures and requirements documents. These checks are carried out by the verifier with the appropriate tools and techniques that have been approved by the IV&V team leader. IV&V reviews released documents that have been independently reviewed by the design team. As is the case with the number of programmers in a project, it is anticipated there will be more than one verifier.

5.4.3.2.3 Librarian

The maintenance of the software library that contains software that has completed the IV&V process is a key element in the IV&V process. The librarian, in the execution of that position, verifies that a project's software conforms to library standards, verifies that software release records provide correct "what-where" information, and communicates library updates to all user groups.

5.4.4 Responsibilities

5.4.4.1 Independent Verification and Validation Team Responsibilities

The IV&V team shall evaluate the software design and test documentation and perform testing.

The emphasis shall be placed on assuring that the documentation detailing the software functional requirements, hardware interface requirements and system performance specifications are clear, accurate and complete.

The documentation shall be reviewed looking for omissions, inconsistencies, inaccuracies and errors of omission/irrelevant requirements. Some significant functional requirements may be identified and monitored as development progresses.

The emphasis shall be placed on full independent analysis of the system requirements and design specifications, as well as on testing and evaluation for the systems requiring the highest reliability.

The actual assignment of team members for engineering, verification, testing, and validation is shown in EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES.

Requirements and the implementation of design shall be evaluated to verify that the resulting system operation is functionally correct and meets the performance objectives.

5.4.5 Tools, Techniques, and Methodologies

5.4.5.1 Automated Tools

Part of the IV&V planning process includes the selection of appropriate tools for a given project.

5.4.5.2 IV&V Core Activities

The following IV&V core activities are applicable to every system.

1. Upon completion of the IV&V review of a particular software item, the reviewer will complete and sign the checklist (Section 13) for the phase in which the preparation of the software item is completed. The questions in this checklist provide a basic set of considerations that the IV&V reviewer shall include in the review.

2. Reviews assure clear, accurate and complete software documentation detailing the design requirements and design specifications.

3. System validation testing, as a minimum, will be performed on an integrated system as part of the development. Hardware functional testing and FAT will be performed as part of the manufacturing processes. Details of test bed, validation test procedures and test results will be documented in accordance with the requirements of Section 5.8.

4. Unit and Module Testing will be performed by the IV&V team according to EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES. The test plan, test procedures and test results will be documented, as required and in accordance with Section 5.8.

5. Commercial off-the-shelf (COTS) software used for Protection and Important-to-Safety class software must go through a Commercial Grade Dedication process prescribed in Reference 4, consistent with the process guidance of Reference 3. A Commercial Grade Dedication report is prepared by the design team. The IV&V team shall review the report to determine its applicability and suitability for meeting the system requirements.

6. If the COTS software to be used for Protection and Important-to-Safety class software has changed since the Commercial Grade Dedication report was issued, then the IV&V team must do one of the following:

    a. Review the changes to COTS software and determine their impact on the system. Evaluate the reported errors for new releases and determine their impact on the application. Revise the Commercial Grade Dedication report including recommended tests to be conducted where an impact is identified.

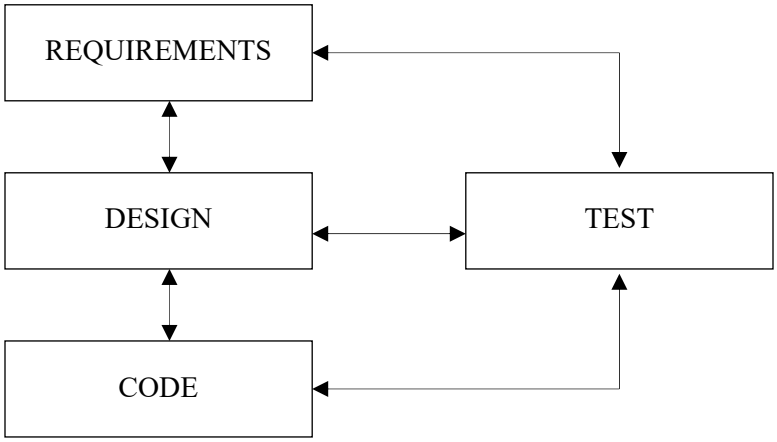
    b. Verify that the changes to the COTS software were performed in accordance with acceptable industry standards (e.g., IEEE 7-4.3.2 [Reference 11] or IEC-60880 [Reference 25]). Revise the Commercial Grade Dedication report.

    Alternatively, these activities can be performed by the design team and reviewed by the IV&V team.

5.4.5.3 Requirements Traceability Analysis

Throughout the software life-cycle, a software requirements traceability analysis (RTA) will be performed and a requirements traceability matrix (RTM) maintained for each system. The design team shall be responsible for the RTM to the point of identifying the code satisfying the requirement, and the IV&V team shall be responsible for adding information to the RTM related to testing that it performs. The IV&V team shall be responsible for the RTA. The IV&V team shall review the RTM for the adequacy and accuracy of the software requirements tracing.

Associating requirements with the documentation and software that satisfy them creates the RTM. The system is verified to show that all applicable requirements have been met. A unique number should identify each requirement. The association between requirements, design, code, and tests can be made using document and section references, test identification numbers, software code identification numbers, etc. The minimum acceptable information to be contained in the RTM is shown in a simple traceability matrix structure below.



The RTM can be either a table of information prepared manually, or a report generated from a requirements database. It is recommended that the RTM be kept in a database format for ease of update, however, the approved version (or generated reports) stored in EDMS shall be the official record.

At the end of the requirements phase, the RTM is first developed from which all subsequent phases will be traced against. After each subsequent phase, the design team shall identify how the requirement is met in that particular phase.

The RTM shall be a living document to be used throughout each phase of the design life cycle process. After each life cycle phase, the design team shall complete the RTM for that phase to verify that all requirements have been properly addressed in that phase. In other words, the design team shall confirm that all lower level requirements and design features are derived from higher level requirements, and that all higher level requirements are allocated to lower level requirements and design features.

Traceability analysis verifies completeness, that all lower level requirements and design features are derived from higher level requirements, and that all higher level requirements are allocated to lower level requirements, design features, and tests. Traceability analysis is also used in managing change and provides the basis for test planning.

The traceability analysis also provides a method to cross-reference each software requirement against all of the documents and other software items in which it is addressed. Requirements entered in the matrix are organized into successive lower level requirements as described in each document. The purpose of this analysis is to verify that the design team addresses every requirement throughout the design life cycle process. The life cycle phases that shall be analyzed are requirements, design, implementation, test and installation/checkout.

The inclusion of revision documents within the analysis shall provide a history of requirements changes throughout the project. Requirements that have been deleted should be indicated by line-out or other means to preserve the historical record.

5.4.5.4 Database Review/Testing

It is not sufficient to test only the algorithm to verify the correctness of a program. It is also necessary to establish the correctness of the database used by that program. This potentially involves review of four different areas by the IV&V team:

- data accuracy
- data completeness
- data structure
- data accessibility

Data accuracy deals with the correctness of the individual data items stored in the database. This is normally verified during software testing; however, the IV&V team may also include a review of data accuracy.

Data completeness verifies that all the data that needs to be present is in fact present in the database. This is normally verified during software testing; however, the IV&V team should review the database to verify that all required fields are present.

Data structure review deals with the analysis of the structure of the database. It may include the ordering of the individual data items within the database as well as the structuring for accurate and efficient searches or access.

Data accessibility reviews determine the extent to which the data items could be modified, intentionally or unintentionally. Methods for "data hiding", that limit the ability to modify data to known software items, are preferred. These methods protect software against unintended function brought on by unexpected changes to data made by unauthorized program functions. In contrast, global data techniques that result in unrestricted access and modification are undesirable.

IV&V database reviews are documented by completing the appropriate sections of EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION DESIGN PHASE CHECKLIST.

5.5 LIFE CYCLE VERIFICATION AND VALIDATION

5.5.1 Management of IV&V

The management of IV&V spans all life-cycle phases. Software development is a cyclic and iterative process. The IV&V effort shall re-perform previous IV&V tasks or initiate new IV&V tasks to address software changes. IV&V tasks are re-performed if errors are discovered in the IV&V inputs or outputs.

Management of IV&V includes:

1. **Software IV&V Plan:** Any deviations or project specific additions to the SVVP shall be defined in the Project Quality Plan (Reference 4) or, in a project specific SVVP. This may include resources and schedule of the specific IV&V activities.

2. **Baseline Change Assessment:** Evaluate proposed software changes for effects on previously completed IV&V tasks. When changes are made, plan iteration of affected tasks which includes re-performing previous IV&V tasks or initiating new IV&V tasks to address the software changes.

3. **Management Review:** Conduct periodic reviews of the IV&V process in the area of technical accomplishments, resource utilization, future planning and risk assessment. Support daily management of IV&V phase activities. Review final and interim IV&V reports. Evaluate IV&V results and anomaly resolution to determine when to proceed to the next life-cycle phase and to define changes to IV&V tasks to improve the process.

4. **Review Support:** Support management and technical reviews (e.g., Software Requirements Review, Architecture Design Review, Critical Design Review, etc.). Identify key review support milestones in SVVP and schedule IV&V tasks to meet milestones. Establish methods to exchange IV&V data and results with design team.

The costs of IV&V shall be identified during the proposal (concept) phase of the project. The resources for performing the IV&V shall be identified in the Project Quality Plan (Reference 4) or project-specific SVVP that is prepared by the Project Manager during the conception phase of the software life cycle.

5.5.2 Concept (Initiation) Phase IV&V

Concept phase IV&V is the period prior to formal definition of the system requirements, which may include a feasibility phase.

Project specific IV&V planning, including schedule and personnel requirements should be developed at this time and incorporated in the Project Quality Plan. Any specific tools to be used must be stated in the plan.

The conceptual design is based on the customer's bid specification, Westinghouse's proposal and the contract.

5.5.2.1 IV&V Inputs

1. Feasibility Study (if applicable)
2. Customer's Bid Specification
3. Westinghouse's Proposal
4. Contract
5. Governing NRC regulations

5.5.2.2 IV&V Tasks

1. Review Concept documents for consistency, incompatibilities, and compliance to regulations
2. Identify major constraints of interfacing systems
3. Identify constraints or limitations of proposed system
4. Assess criticality of each software item
5. Configuration management evaluation of all applicable conceptual documents (including evaluating if conceptual documents have been captured properly and placed under configuration control).
6. Verify tracing of project baseline documents for compliance to customer requirements, applicable product documents and regulatory standards and guidelines.
7. Complete EXHIBIT 5-2 CHECKLIST NO. 1, SOFTWARE VERIFICATION AND VALIDATION CONCEPT PHASE CHECKLIST

5.5.2.3 IV&V Outputs

Reporting of the concept review activities can be incorporated in the Requirements Phase report, including identification of deficiencies, and the completed EXHIBIT 5-2 CHECKLIST NO. 1, SOFTWARE VERIFICATION AND VALIDATION CONCEPT PHASE CHECKLIST.

5.5.3 Requirements Phase IV&V

The intent of verifying the system (or functional) requirements is to ascertain that the requirements are complete, correct, consistent, clear, traceable, and testable. The main purpose of this system requirements review is for the designer to understand the requirements.

The system requirements form the basis of all the system design and verification efforts, and are used throughout the rest of the product life cycle. They serve as the basis for the verification of design specifications, which, in turn, are the basis for the verification of design implementation. System requirements are the bases against which all the validation activities are performed.

The principal purpose of a requirements document is:

1. To clearly define the objectives and needs of the system design and development process. Both the designer and the user must be able to understand and perform a meaningful assessment of the system.

2. To serve as a means against which an implementation can be validated and the intermediate steps can be verified.

The goal of verification activities during this phase is to confirm that the requirements documents do indeed serve the above purpose.

In order to satisfy the need of both the IV&V and designer to understand and evaluate the system, real-time system requirements should be stated in clear, concise, and understandable terms. Extraneous issues, which are not requirements, should not be in the System Requirements Specification (SysRS) or it should be explicitly stated that they are for information only.

As a common practice, complex systems are systematically decomposed into smaller subsystems and their functions are assigned to either hardware or software. In some systems, in order to present a clear picture, decomposition may include data flow, control flow, and intricate synchronization and timing aspects and implicitly specify the software and hardware architectural requirements.

5.5.3.1 IV&V Inputs

1. System Requirements Specifications
2. Interface Requirements Documents
3. Existing User documentation
4. Requirements Traceability Matrix
5. Other documented requirements, such as:

    a. Design inputs
    b. Functional diagrams wiring, diagrams, etc.
    c. Historical design, test and development records
    d. Instrument configuration documents
    e. Acceptance test documents
    f. Qualification test reports

The Interface requirements document(s) should not be generated unless it is an explicit project requirement. The interface information can be stated in the SysRS. The fewer the sources of requirements the less chance of error in creating and reviewing these requirements.

5.5.3.2 IV&V Tasks

The major objectives of the verification activities during this phase are to:

1. Evaluate the adequacy of the allocation of system requirements to hardware, software, and subsystems.

2. Evaluate the feasibility of accomplishing the system objectives and goals with the assigned requirements and using the allotted processor resources

3. Verify design requirements are complete, accurate, testable, and unambiguous as possible

4. Perform software safety requirements analysis review

    a. Verify identification of any hazards and software safety requirements

    b. Verify identification of any software safety design constraints and guidelines

    c. Verify identification of any software safety test requirements and provide inputs to the test planning process

d. Verify identification of any required, encouraged, discouraged and forbidden design, coding and test techniques

Verifying the system architecture and decomposition is one of the IV&V tasks. The IV&V team reviews the interrelationship between hardware/software and subsystems to verify that the overall integrated system does indeed have potential to meet the system needs and objectives. The following are specific IV&V Tasks:

1. Review the adequacy and accuracy of the Requirements Traceability Matrix (RTM) as prepared by the design team. The traceability in the RTM is established in both directions at each decomposition level and allows IV&V to verify the software requirements are complete, correct, and accurate decomposition of allocated system requirements. The review shall include verification that all functional, hardware interface, software, performance, and user requirements have been included.

2. Assess allocation of functions to hardware and software items

3. Perform or review the adequacy and accuracy of the following software safety analyses using Reference 26, Annex A.1 as criteria:

   a. Criticality
   b. Specification
   c. Timing and sizing
   d. Different software system (if applicable)

4. Complete EXHIBIT 5-3 CHECKLIST NO. 2, SOFTWARE VERIFICATION AND VALIDATION REQUIREMENTS PHASE CHECKLIST.

5. Other IV&V review areas should include:

   a. Review requirements source documents - what is the basis of the requirements?

   b. Review system requirements - does the system design implement the functional requirements, are the plant parameters defined in the functional design being monitored in the system design?

   c. Perform analysis of requirements decomposition - are subsystems defined with interface requirements noted?

   d. Review test requirements - what testing is needed and how will it be judged (i.e., what are the acceptance criteria)?

e. Review data interface requirements - are data management requirements consistent with hardware requirements?

f. Review human factors requirements – ease of interaction of the system with operation, maintenance, and testing.

6. Review requirements with respect to possible errors. See EXHIBIT 5-3 CHECKLIST NO. 2, SOFTWARE VERIFICATION AND VALIDATION REQUIREMENTS PHASE CHECKLIST for a detailed list of possible errors.

7. Tools used in the development process (such as computers) do not require IV&V as long as the resultant code is subject to IV&V. Configuration management of these tools will be under the Software Configuration management plan Section 6.

8. The Design team reviews previously developed or sub-vendor software in the following areas and produces a Commercial Grade Dedication Report stating whether this software is adequate for its intended use. The IV&V team reviews the Commercial Grade Dedication Report to evaluate the suitability of the commercially dedicated item for the particular implementation being verified.

   a. The software used and its documentation shall be maintained and controlled during development, implementation, and testing. Procedures shall state how verification of the configuration is to be accomplished to assure that the software used for testing is the same as that used for the final system.

   b. The software and its use shall be described in sufficient detail for an independent verification to determine the impact of using this software. This description would include the following:

      1) Adequacy of the documentation (complete, unambiguous, and consistent with the software)

      2) User interface with the software

      3) Use of the software in development

      4) What control the software has over the final output; e.g., is the software primarily used as a documentation tool or does it influence the exact software running in the delivered system

      5) A description of how the software will be changed after installation; or if a tool, will be used to make change

6) User documentation

7) Test plans and test cases used to validate the software for acceptability

c. A method of notifying the user if errors are discovered in use of this program after installation which may affect operation

d. A determination of what, if any, additional documentation, testing, or reviews are required to validate the use of this software in the system development

e. The software and its use shall be included in the Software Hazards analysis for the Common Q™ System in which it is used

9. Verify identification of the original software items developed under this SPM for generic application that will be used in the project; verify that the qualification status has been identified and is appropriate; and verify through the RTA process that this software meets the requirements.

10. Develop a Common Q™ specific test plan in accordance with the requirements in subsection 4.3.2.2.

11. Configuration Management Evaluation – assess the applicability of the Software Configuration Management Plan (Section 6) to the project as augmented by the project plan.

12. A review shall be conducted to verify that each hazard identified in the software hazard analysis and/or failure modes and effects analysis, has been mitigated or the risks associated with the hazard have been reduced to an acceptable level.

The IV&V team may obtain the documentation required from the supplier or perform a documented review of the documentation at the supplier facility to determine acceptability. The installed base of software installed and operating in similar environments and also vendor records of changes repair may be considered by the IV&V team in their review.

If the IV&V team review of this software finds it acceptable, the IV&V team shall verify that the Certificate of Conformance to be issued (if required by contract) when the system ships to the client, certifies that the procured software (name, manufacturer, part/model number, revision) is acceptable for use.

5.5.3.3 IV&V Outputs

1. Completed EXHIBIT 5-3 CHECKLIST NO. 2, SOFTWARE VERIFICATION AND VALIDATION REQUIREMENTS PHASE CHECKLIST.

2. Produce a report on concept and requirements review activities, including identification of deficiencies.

3. Test Plan in accordance with subsection 4.3.2.2.

5.5.4 Design Phase IV&V

The purpose of design specification verification is to ascertain that the design specifications are a faithful translation of the design requirements before the design is committed for implementation.

The design specification documents define and provide the details of the system design structure, information flow, processing steps and other aspects required to be implemented, in order to satisfy the system design requirements. The intent of the design specification verification is to verify that the design specifications are clear and understandable, accurate, correct, consistent, complete, implementable, testable, and traceable to the design requirements.

Considering the inherent iterative nature of design activities, IV&V tasks are conducted on an ongoing basis. This is highly desirable especially when IV&V efforts parallel design activities. Test planning and verifying the conformance of design documentation to established standards are the major objectives of preliminary IV&V activities. As the design progresses, the design as documented is analyzed and critically evaluated for its potential to meet design requirements.

5.5.4.1 IV&V Inputs

1. Design documentation, including (as necessary for the project scope):

    a. Hardware design specification(s) (as it relates to the software interface)
    b. Software design description(s)
    c. Interface design specifications

2. Requirements documentation from the previous phase
3. Other standards and requirements
4. Requirements Traceability Matrix

5.5.4.2 IV&V Tasks

1. Review system design documentation to verify the system design completely and correctly performs the functions specified in the requirements documents

2. Review system design documentation to determine that the hardware/software interface design specifications are understandable, unambiguous, reasonable, implementable, accurate, complete, and are a faithful translation of the hardware/software interface design requirements into hardware/software interface design specifications

3. Review software design documentation to verify design requirements are adequately incorporated. The design documentation shall address all software requirements and provide a correlation of the design elements with the software requirements.

4. Perform or review the adequacy and accuracy of the following software safety design analyses using Reference 26, Annex A.2 as criteria:

   a. Logic
   b. Data
   c. Interface
   d. Constraint
   e. Functional
   f. Software element

5. Review current criticality analysis assessment for continued applicability.

6. Complete EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION DESIGN PHASE CHECKLIST.

7. Perform the Requirements Traceability Analysis.

8. Configuration Management – Confirm that the verified design documents have been properly placed under configuration control.

9. Begin preparing module, unit, integration, system validation and FAT test procedures in accordance with Section 5.8.

10. Review the software hazard analysis and/or failure modes and effects analysis to verify that any new hazards have been documented during this phase.

5.5.4.3 IV&V Outputs

1. Completed EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION DESIGN PHASE CHECKLIST

2. Produce a report on the design review activity, including identification of deficiencies and possible enhancements

3. Follow-up as changes and corrections are incorporated into the requirements

5.5.5 Implementation Phase IV&V

The purpose of the implementation verification is to ascertain the implementation documents are clear, understandable, logically correct and a faithful translation of the design specifications. The objectives of the implementation documents are to facilitate the effective production, testing, use, transfer, conversion to a different environment, future modifications, and traceability to design specifications. In general the verification activities during this phase are oriented towards evaluating the following:

1. Does the implementation satisfy design specifications?

2. Does the implementation follow established design standards?

3. Does the implementation follow established documentation standards?

4. Does the implementation serve production, test, use, transfer and other needs that motivated its creation?

5. What is involved in testing the actual resulting product?

5.5.5.1 IV&V Inputs

1. Software/Hardware design documents
2. Source code and executable code
3. Interface design documentation
4. Other standards and procedures
5. Software Configuration Management Procedures
6. Module Test Reports
7. Requirements Traceability Matrix

5.5.5.2 IV&V Tasks

1. The IV&V team shall review the as-built software documentation to verify the as-built software completely and correctly implements the design specified in the system design documents

2. Perform or review the adequacy and accuracy of the following software safety code analyses using Reference 26, Annex A.3 as criteria:

   a. Logic
   b. Data
   c. Interface
   d. Constraint
   e. Programming style
   f. Non-critical code
   g. Timing and sizing

3. Review current criticality analysis assessment for continued applicability.

4. Review module test reports (if applicable) and unit test reports, and verify correct execution of critical software elements. Complete the applicable section of EXHIBIT 5-5 CHECKLIST NO. 4, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST.

5. Review the code and associated database(s) for complete and correct implementation of the design. Complete the applicable sections of EXHIBIT 5-5 CHECKLIST NO. 4, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST.

6. Review the hardware/configured software integration procedure to verify they are complete and correct.

7. Perform the Requirements Traceability Analysis

8. Evaluate Software Configuration Management activities and verify the requirements of Section 6 are fulfilled

9. Hardware implementation review is normally conducted as part of the hardware quality assurance activities defined elsewhere

10. For protection class software, review software testing records to verify adequate structural testing[1]

11. Integration, System Validation and Factory Acceptance test procedures shall be prepared in accordance with Section 5.8, based upon the requirements of the design and shall include test

1. Structural testing is testing that validates all branches of a software module.

cases encompassing the range of usage intended for the system. The tests shall specify the following, as applicable:

a. Identification of the test cases.

b. Description of the test cases.

c. Relationship of the test cases with the requirements, both functional and safety, and testing of all applicable program logic.

d. Expected results of the test cases with acceptance criteria.

e. Special requirements or conditions for the test, such as hardware configuration, monitoring hardware or software, sequencing of tests, etc.

f. The simulation of the inputs shall be documented, including any special hardware or software required for these simulations.

g. Procedures to report errors found during testing, and acceptable means of retesting these errors after error correction has been performed.

h. The validation test procedures shall address the following questions:

    1) Is the test procedure description complete?

    2) Are the test problem definitions adequate and complete?

    3) Is each testable requirement adequately covered?

    4) Is the plan for evaluating and reporting test results adequate?

12. Review the software hazard analysis and/or failure modes and effects analysis to verify that any new hazards have been documented during this phase.

5.5.5.3 IV&V Outputs

1. Software Module Test.

2. Completed EXHIBIT 5-5 CHECKLIST NO. 4, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST.

3. Produce a summary report on Implementation Review activity, including identification of deficiencies and possible enhancements.

4. Follow-up as changes and corrections are incorporated into the implementation.

5. Test Procedures.

5.5.6 Test Phase IV&V

The verification process has provided an orderly step-by-step assurance of a true translation through the requirements, design, and implementation phases, each step being assessed upon the basis of the previous step. The integration and system validation process involves determining whether the system meets its functional requirements; e.g., functional operations, system level performance, external interfaces, internal interfaces, testability, and other requirements as stated during the definition phase. Integration and System validation evaluates the system performance in an environment that is real, or as close to real as can reasonably be created; therefore, the fully integrated system with the actual system hardware and software is required. In large system applications, it may be required that validation testing begins at the subsystem level. Subsystem validation is usually desirable, to ease the error/failure isolation, even if not mandated.

The validation test environment must be configured to fit the system being tested. It should be matched to the available resources as much as practical to create the real operating environment.

The integration and system validation process includes a Software Safety Test Analysis that demonstrates that safety requirements have been correctly implemented and the software functions safely within its specified environment. This analysis is documented by completing EXHIBIT 5-6 CHECKLIST NO. 5, SOFTWARE VERIFICATION AND VALIDATION TEST PHASE CHECKLIST. In some instances, system validation activities overlap those conducted earlier during verification and/or subsystem validation. Typical validation tasks are listed below:

1. The system functional operation is validated using the "black box" method; i.e., validating the system outputs by means of actuating prescribed inputs. Validation is conducted using the limits and ranges as designated in the system functional requirements, which are included in the system design requirements. The major validation areas shall be:

    a. Functional operation

    b. System level performance – demonstrates software's performance within overall system

    c. External and internal interfaces – demonstrating that critical computer software units execute together as specified

d. Stress testing – demonstrates that the software will not cause hazards under abnormal circumstances

e. Regression testing – demonstrates changes made to the software do not introduce conditions for new hazards or errors

2. Failure performance testing is executed on a functional operations basis.

3. Transient tests are executed to validate system functional operations.

4. Integration, System validation and FAT procedures are updated if required.

5. Final developer's documentation, to be:

   a. Complete,
   b. Accurate/compatible with delivered system, and
   c. Compliant with standards.

6. Validation test results are evaluated to be:

   a. Complete/consistent with procedures,
   b. Traceable to functional requirements, and
   c. Document results in test report

5.5.6.1 IV&V Inputs

1. Source code
2. Executable code
3. Applicable library routines
4. User documentation
5. Code analysis tools
6. Hardware environment as close to the installation configuration as possible
7. Requirements Traceability Matrix

5.5.6.2 IV&V Tasks

1. Verify program integration with the deliverable hardware per EXHIBIT 5-6 CHECKLIST NO. 5, SOFTWARE VERIFICATION AND VALIDATION TEST PHASE CHECKLIST to verify that all aspects have been considered.

2. Perform validation testing (integration and system validation testing) in accordance with approved test procedures.

3. The validation test(s) shall be documented in a report. The report can consist of a completed copy of the test procedure form with all blank information completed. The report shall include the following, as applicable:

   a. Computer software/PROM version tested

   b. Configuration of all hardware used (model number/serial number)

   c. Test equipment used and calibration data, if applicable

   d. Date of test and personnel performing the test

   e. Test problems

   f. Results and acceptability

   g. Action taken in connection with any deviations noted. Errors and their correction shall be documented and IV&V'd in parallel with change control procedures found in Section 6.

   The validation test report(s) shall address the following questions:

   a. Do the test results comply with the format specified in the test procedure?
   b. Do the test results provide an accurate statement of the testing performed?
   c. Are the test results acceptable and auditable by persons not involved with the test?

   Documentation of these reviews shall consist of completing EXHIBIT 5-6 CHECKLIST NO. 5, SOFTWARE VERIFICATION AND VALIDATION TEST PHASE CHECKLIST.

4. Follow up on changes and corrections made in the system in accordance with change control procedures in Section 6.

5. Perform the Requirements Traceability Analysis.

6. Review user documentation. This may be done as part of the Installation and Checkout phase if within Westinghouse's scope of supply by specific contract.

7. Perform Functional Review to verify that all requirements specified in the SRS have been met. This review shall include an overview of all documentation and a review of the results of the previous reviews, including Software Requirements Review, ADR, CDR, and if applicable,

interim IV&V reports (for Protection and Important-to-Safety class software). The tasks conducted in this phase meet the requirements of subsection 4.6.2.5, Functional Review.

8. At the completion of all other tasks listed above, a final IV&V report is issued. The final IV&V report may not be issued until the Installation and Checkout Phase if within Westinghouse's scope of supply by specific contract.

5.5.6.3 IV&V Outputs

1. Test Report and evaluation for acceptability

2. Completed EXHIBIT 5-6 CHECKLIST NO. 5, SOFTWARE VERIFICATION AND VALIDATION TEST PHASE CHECKLIST.

3. Produce a summary report on test phase IV&V activity results, including identification of deficiencies and possible enhancements.

4. Code certificates certifying that the software is acceptable for use.

5.5.7 Installation and Checkout Phase IV&V

If within Westinghouse's scope of supply, the system installation package shall be reviewed to verify that all elements necessary to install and operate the system have been correctly and completely specified.

5.5.7.1 IV&V Inputs

1. Installation procedures, system generation procedures, etc.
2. User documentation

5.5.7.2 IV&V Tasks

1. Review installation procedures and user manuals to verify that they are complete and correct
2. Review training materials (if within Westinghouse's scope of supply) for the following:

    a. Safety training for the users, operators, maintenance and management personnel
    b. System startup training
    c. Safety training requirements are met

3. Review the Exception Report Log that was maintained in accordance with the SAT plan.

4. Prepare and issue the final IV&V report. This report will be issued at the conclusion of the Test Phase if the Installation and Checkout Phase are not within Westinghouse's scope of supply. This report provides:

    a. A listing of all IV&V documentation produced. This documentation shall include records of the following reviews as a minimum: Hardware interface requirements review; Software design requirements review; Audit results of previously-developed software; Configuration implementation review; Hardware/configured software integration review (if separate from validation testing); Test procedure/test report review; and Installation/checkout review. All reviews shall be conducted in a similar manner and at least have the following format (as a minimum):

        1) Review summary
        2) Recommendations (including any requirements for further reviews)
        3) Detailed review comments and resultant actions

    b. A listing of deficiencies detected with corrective action taken.

    c. An evaluation of the system based upon the IV&V.

    d. Comments and recommendations to aid in future system upgrades and development.

5. Complete EXHIBIT 5-7 CHECKLIST NO. 6, SOFTWARE VERIFICATION AND VALIDATION INSTALLATION AND CHECKOUT PHASE CHECKLIST.

6. Configuration Management – Evaluate that the manuals and procedures have been properly placed under configuration control.

5.5.7.3 IV&V Outputs

1. Final IV&V report (if within Westinghouse's scope of supply) with summary review of the system's acceptability.

2. Completed EXHIBIT 5-7 CHECKLIST NO. 6, SOFTWARE VERIFICATION AND VALIDATION INSTALLATION AND CHECKOUT PHASE CHECKLIST.

5.5.8 Operation and Maintenance Phase IV&V

Situations may arise after installation of an IV&V'd computer system, which may require the performance of additional IV&V activities:

- Modifications are made in the hardware, which may cause the software to be changed.
- Modifications are made to the program for enhancements.
- Errors may be discovered which require software modifications.

The IV&V activities required for program modifications are identical to those previously discussed for new program development. However, if the program modification is such that it does not affect some phase of the IV&V (for example, a code error might not affect the system requirements or design documentation), these areas of IV&V may be omitted.

During this phase, IV&V shall evaluate the new system or software requirements to verify the applicability of this SVVP. Any necessary changes to the SVVP shall be documented in the Project Plan for the modification.

An IV&V report shall document all IV&V activities regarding the modification. This must include, or reference, a regression analysis including test requirements and results.

A new code certificate must be prepared that references the original IV&V report, and the final IV&V report for the modification.

5.6 SOFTWARE VERIFICATION AND VALIDATION REPORTING

IV&V reporting shall occur throughout the entire software life cycle and include the following (which have been identified in the software life cycle activities).

5.6.1 Required Reports

1. IV&V phase summary reports: These reports are issued after each life cycle phase of the IV&V task to summarize the IV&V review. Phase summary reports may be consolidated into a single report if desired. These reports shall contain the following:

    a. Description of IV&V tasks performed
    b. Summary of task results
    c. Summary of discrepancies and their resolution
    d. Assessment of software quality
    e. Recommendations

2. Discrepancy reports: These reports must be consistent with EXHIBIT 11-1 EXCEPTION REPORT. These reports shall document each discrepancy found during the IV&V reviews and include:

a. Title, number, and revision of document reviewed.
b. Section/Page reference location
c. IV&V comment
d. Resolution with design team

3. Final IV&V Report: This report shall be issued at the end of the IV&V task to summarize and document the IV&V activities performed throughout all life cycle phases. The report shall include:

a. Summary of life cycle IV&V tasks
b. Summary of task results
c. Summary of discrepancies found and resolutions
d. Assessment of overall software and system quality
e. Recommendations for enhancements
f. Code certificate

5.6.2 Optional Reports

Other reports may be produced as required to document special hardware testing activities, human factors reviews, etc. The format of these reports shall include purpose, approach, and summary of results as a minimum.

5.7 VERIFICATION AND VALIDATION ADMINISTRATIVE PROCEDURES

5.7.1 Anomaly Reporting and Resolution

Any discrepancies detected during any phase of the IV&V process should be immediately brought to the attention of the design team and the Project Manager of the development. Resolution shall be made in writing by the design team. The IV&V team must document the resolution in the IV&V phase summary reports as well as the final IV&V report.

5.7.2 Task Iteration Policy

If the IV&V task must be re-performed, for whatever reason, the task must be identified in the reports produced identifying the rationale and the results of the IV&V task. This information should be documented in a Revision Abstract for revised IV&V reports, unless a separate regression analysis document is issued in lieu of a revised IV&V report.

5.7.3 Deviation Policy

If any deviation is planned from the reviewed and approved IV&V task plan, the change must be identified, rationale for the change provided, and a determination of effect on software quality provided. Any deviation must be documented in a Project Quality Plan and approved by the IV&V team leader and management.

5.7.4 Control Procedures

Procedures in this Software Program Manual (and those generated for specific Common Q™ subsystems as directed by this manual) for IV&V and software development provide the controls for the activities associated with these efforts.

5.7.5 Standards, Practices, and Conventions

Specific standards, practices, and conventions for the IV&V effort which differ from those stated in this procedure and its references shall be specifically stated in the project specific Project Quality Plan.

5.8 IV&V TEST DOCUMENTATION REQUIREMENTS

The purpose of this section is to define the purpose, format and content of required test documentation. The test documentation as a whole shall fulfill the requirements of References 14 and 20.

5.8.1 Test Plan

The test plan documents the scope, approach, resources, and schedule for the testing activities of the project. It identifies the test items, the method for identifying the specific requirements to be tested, the testing tasks, and the required resources to perform these tasks. Subsection 4.3.2.2 contains the requirements for the test plan. See Section 7 for the Common Q™ testing methodology.

5.8.2 Test Procedure

The elements of the test specification and test cases described in Reference 14 can be found in the test procedure. The test procedure shall comply with the requirements of Reference 14, Section 7.

5.8.2.1 Test-Design Specification

This portion of the test procedure specifies the details of the test approach for a software requirement or combination of requirements, and identifies the associated tests.

5.8.2.2 Test-Case Specification

This portion of the test procedure specifies the inputs, predicted results and a set of conditions for executing the test case.

5.8.2.3 Test-Procedure Specification

This portion of the test procedure specifies a sequence of actions for the execution of a test.

5.8.3 Test Report

The test report summarizes the testing activities, and documents the results. It also contains an evaluation of the corresponding test items. Typically the test procedure document containing the hand-written entries by the tester becomes a part of the document.

The test report also contains the Exception Report log and copies of the Exception Reports. Together, these identify the status of outstanding test exceptions reported during testing. The test reports shall comply with the requirements of Reference 14, Section 11.

5.9 SOFTWARE INTEGRITY LEVEL SCHEME

There is not a direct correlation between the software integrity levels in IEEE Std. 1012-2004 and the software classification described in Section 1 of this Software Program Manual. For software items not classified in EXHIBIT 4-1 ASSIGNMENT OF COMMON Q™ SOFTWARE TO CLASSES, a Safety Classification Record (Reference 4) shall describe the agreed upon software classifications established for the system. The mapping of the software classifications in this manual to those of the IEEE Std. 1012-2004 is as follows:

Table 5.9-1. Software Classification Mapping

SPM Classification	IEEE Standard 1012-2004
Protection	4
Important-to-Safety	4 (with noted exceptions identified in EXHIBIT 5-8 IEEE STANDARD 1012-2004 COMPLIANCE TABLE)
Important-to-Availability	2 – See EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES
General Purpose	1 – See EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES

(Last Page of Section 5)

SECTION 6
SOFTWARE CONFIGURATION MANAGEMENT PLAN

6.1 INTRODUCTION

6.1.1 Purpose

Software Configuration Management (SCM) is the process for identifying software configuration items, controlling the implementation and changes to software, recording and reporting the status of changes, and verifying the completeness and correctness of the released software. SCM is intended to be utilized throughout the entire software life cycle, including requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase, and retirement phase.

The intent of this document is to provide additional guidance and recommendations on employing SCM for Common Q™ software systems, and to adhere to industry guidelines on SCM defined in the Reference documents. This plan conforms to the requirements of U.S. NRC Regulatory Guide 1.169, "Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Plants," July, 2013 (Reference 19), for configuration management plans. This SCM Plan conforms to the requirements of Reference 10.

This document will also provide recommendations on the level of SCM required for various types of software development projects. When it is necessary for an individual software development effort to differ from these guidelines or add additional requirements, the Project Quality Plan (Reference 4) should incorporate these changes or a separate configuration management plan may be developed.

The goals of software configuration management are to:

1. Record and document work in progress on each software item to permit understanding of current project status.

2. Identify all software code and data associated with a system including revision level, completion status, test status and history.

3. Maintain the association among software documents, code, and data.

4. Identify sets of software items that compose the system (baseline), test status and history, and readiness for release.

5. Maintain the status of released software, users of this software, and associated exception reports.

6. Maintain an association between software errors, change reports, and affected documentation, code, and data items.

7. Implement appropriate controls and approvals for changes to the software configuration.

8. Identify the organization responsible for a software item and its associated exception reports and changes.

9. Document criteria for generation of software to release for use.

10. Provide the means for existing and prior revisions of software to be reconstituted in the future.

11. Backup the software (in progress or completed) to protect against disaster.

12. Plan for controlling access to software and protecting against software viruses.

6.1.2 Scope

SCM shall be applied to all Common Q™ software and software tools used in the development of Common Q™ software. Software intended for limited use, such as in a single design analysis, may be used without employing SCM provided that the results as well as method and/or formulas are documented in the design analysis in sufficient detail to allow independent verification. An example of this is the use of Microsoft Excel to develop a design calculation.

All software items and associated documentation shall be controlled in such a manner as to maintain the items in a known and consistent state at all times. New software and modifications to existing software shall follow the configuration requirements for all life cycle phases. Existing software that is not to be modified, including tools used in the software development, test, and documentation process, shall fall under these configuration control procedures upon modification.

SCM shall be applied to software in any form, including (but not limited to):

1. magnetic tapes
2. magnetic disks
3. magnetic diskettes
4. optical disks and diskettes
5. non-alterable devices such as Read Only Memories (ROMs) alterable devices such as Programmable Read Only Memories (PROMs), Electrically Alterable Read Only Memories (EAROMs), Electrically Programmable Read Only Memories (EPROMs), etc.

Documentation of the software, such as listings, drawings, specifications, etc., shall also be subject to configuration management in accordance with procedures for document and drawings control defined in Reference 4.

6.1.3 Definitions

Refer to page xiii for a list of acronyms and trademarks. Refer to page xvi for definitions.

6.1.4 References

Refer to page xx for a list of references.

6.2 MANAGEMENT

6.2.1 Organization

All software configuration management functions for a system are performed in accordance with Reference 4 by the NA organization. IV&V activities related to configuration management are performed by member(s) of the IV&V team.

6.2.2 SCM Responsibilities

The design team and the IV&V Group in the Nuclear Automation organization are responsible for implementation of adequate measures to manage and control the software configuration of a Common Q™ project during all phases of the software life cycle.

Specific SCM responsibilities are defined below in accordance with the software life cycle phases.

6.2.2.1 Requirement Phase

1. Identify original software items developed under this SPM for generic application that are to be controlled via this SCMP; assure the qualification of these items are complete and appropriate for the project (including appropriateness of software classification); and describe in the project plan how this software will be integrated with the project-specific software development in terms of producing an RTM.

2. Place requirements documentation under configuration control before submittal to the IV&V team for review. Requirements documentation includes the System Requirements Specification (SysRS) and the Software Requirements Specification (SRS).

3. Establish organizational responsibility for SCM activities. For large projects, a software librarian and/or system administrator may be named to perform the following activities:

   a. Maintain controlled software

   b. Maintain records

   c. Maintain backup copies of the deliverable software in a separate building for security and hazards prevention

   d. Maintain backup copies of software tools used in development, integration, and testing

6.2.2.2 Design Phase

1. Place design documentation under configuration control before submittal to the IV&V team for review. Design documentation includes the Software Design Description (SDD).

6.2.2.3 Implementation Phase

1. Define software items that are to be controlled via this SCMP.

2. Place test plans under configuration control.

3. Software shall be entered into a controlled access account when the programmer is satisfied with the quality of the software and prior to formal testing. System testing is conducted from this controlled access account. The IV&V team shall control the test system hardware/software configuration.

4. Place module test procedures and module test reports under configuration control.

5. Place unit test procedures and unit test reports under configuration control.

6.2.2.4 Test Phase

1. Freeze software/hardware configuration, and document this configuration in the test procedure(s). This configuration then becomes the baseline.

2. Place integration test procedures and integration test reports under configuration control.

3. Place system validation test procedures, FAT procedures, and system test reports under configuration control.

4. Maintain the Exception Report database to track anomalies.

5. Maintain the Software Change Request (SCR) database to track software changes or required enhancements. An SCR may be used to close several Exception Reports.

6. Document final software configuration in the test report and (if required) the IV&V report.

7. Place user documentation under configuration control before submittal to the IV&V team for review. User documentation includes installation procedures, system generation procedures, and system maintenance information. User documentation is normally provided in a Technical Manual. User documentation is reviewed by IV&V during the Installation and Checkout Phase if within Westinghouse's scope of supply.

8. Place the Verification and Validation Report and Computer Code Certificate under configuration control.

6.2.2.5 Installation and Checkout Phase

1. Place installation test procedures and installation test reports under configuration control.

2. Confirm that all As-Built documentation is under configuration control.

6.2.2.6 Operations and Maintenance Phase

1. Document errors found by design engineering and by the user using the information required by EXHIBIT 11-1 EXCEPTION REPORT.

2. Control software changes made by design engineering using SCM procedures.

3. Maintain the Configuration Status Accounting of the delivered software. This includes information on the status of documentation, software items, Exception Reports, Software Release Records and error notifications.

4. Use Software Release Records to identify recipients of any Technical Bulletins required for software error notification.

5. The Platform Lead reviews sub-vendor software problem reports for sub-vendor software used in the delivered system to determine if any are applicable. If applicable, the problem should be identified to users of the software by issuing a Technical Bulletin (Reference 4). The Platform Lead is also responsible for software changes required to correct this error using the SCM procedures.

6.2.2.7 Retirement Phase

1. Software items that are no longer supported by Westinghouse enter the retirement phase of the software life cycle. The Platform Lead should notify users of all software items that have entered the retirement phase. Notification is accomplished by issuing a Technical Bulletin in accordance with Reference 4.

6.2.2.8 Configuration Identification Management

The EPM responsible for the software item(s) is responsible for identification of all separately identifiable modules comprising the software item(s) in any form along with any required documentation.

6.2.2.9 Configuration Control Management

The IV&V group ELM and design group ELM, or designee, are responsible for management of SCM activities.

6.2.2.10 Configuration Status Accounting Management

The IV&V group ELM, or designee, is responsible for collecting data and reporting of SCM activities to the design team, to external groups, and to the end user.

6.2.2.11 Configuration Reviews and Audits

The EPM is responsible to coordinate technical reviews within and external to the project team. Audits by the Quality organization are coordinated through the EPM or ELM. External technical audits/reviews are coordinated through the EPM. External quality audits are coordinated through the Quality organization in conjunction with the EPM.

6.2.2.12 Configuration Control Board

The Configuration Control Board (CCB) shall meet periodically. The CCB shall have the following objectives and responsibilities:

- Review and approve standard (generic) Software Change Requests (SCRs)
- Identify what resources are required to make software changes
- Drive future projects to take advantage of existing generic software/libraries
- Review the progress and status of open SCRs
- Review and approve changes to Common Q™ process documents

A CCB chairperson shall be appointed. Other roles that are also part of the CCB are:

- Design group ELM
- Lead engineers of existing software projects
- Platform engineers

The agenda for the CCB meetings shall be documented.

6.2.3 Applicable Policies, Directives, and Procedures

The requirements of Reference 4 apply and take precedence to these procedures for all Common Q™ software.

Requirements for documentation and drawings control are found in Reference 4.

6.2.4 Management of the SCM Process

The anticipated software development cost includes the SCM process costs, and is detailed in the PQP. The IV&V team performs independent surveillance of SCM activities to verify compliance with the SCM Plan, as defined in subsection 4.6.2.9 of the SQAP. Any risks associated with the SCMP are identified in the PQP.

6.3 SOFTWARE CONFIGURATION MANAGEMENT ACTIVITIES

6.3.1 Configuration Identification

All software (including firmware and ROM code) and documentation shall be uniquely identified. The identification structure shall also have the ability to track errors, resolution of errors, and software items that comprise a system or subsystem.

1. Documentation shall be identified and controlled in accordance with Reference 4.

2. Drawings shall be identified and controlled in accordance with Reference 4.

3. Software shall be identified in accordance with the following requirements, which depend on the format of the software.

   Source and object files for software items must be identified by a unique name, a unique number, and a revision number. For example, object files may be identified by a date time stamp. The EPM shall have the responsibility for defining the name/numbering system for a project. If the project specific SCM plan does not define software identification requirements, the following shall be utilized:

   Source File for Westinghouse created Flat Panel Display Software and Custom PC Element Software – The source file shall contain a program header block that includes the following information:

   - Module name
   - Two-level version identification including successive versions, which implement revised software requirements and correct errors in the code that do not require changes to the software requirements

   The header block shall contain a complete revision history of the software item, including comments on each version and revision. In addition, the header block shall contain the following information:

   - Version information (VV-RR)
   - Programmer
   - Brief description of the program
   - Date
   - Other information as necessary in a comment field

   For example, a typical header block in a source file might contain:

   Module CALCBLOC-00-01

   Revision 00-02

   Control Algorithm Calculation Subroutine

   Copyright notice

   Description: This program calculates control algorithm setpoint offset values from entered user input of setpoints.

   Revision History:

Version:	Author:	Date;	Comments:
00-02	H. Kim	07-Jan-94	This revision implements SCR number SCR-2000000-018 to correct roundoff errors. It also corrects internal naming conventions and adds additional comment fields.
00-01	H. Kim	14-Dec-93	Baseline Version

AC160 Function Chart Type Circuits and Application Programs – Only the name and version/revision of the type circuit or application program is in the function chart diagram.

4. Media – The physical item containing software items shall be labeled with a standard convention and include the following information:

   - Name of the software configuration item
   - Version information

5. Software System – The collection of modules (object files, data files, etc.) representing the entire software for a product which may contain more than one computer is identified at the time of project baseline and updated for all changes to the software contained within. This shall be in the form of a list, which is identified in the Integration Test, System Validation Test, and FAT Reports. This list shall contain the media the software is contained on and an overall product version number. Media identification shall also be provided. The following list is an example:

   Common Q™ HJTC

End User:	Utility
Product Version/Revision:	01/05

   This list should also be on the Computer Code Certificate or may be attached to it (with indication that it is a multi-page Code Certificate).

   Vendor proprietary software identification schemes and labeling shall be defined in the commercial grade dedication report for that software.

6.3.1.1 Acquiring Configuration Items

The process for placing code, documentation, and the data of identified baselines into controlled software libraries is defined in Section 4.11 "Media Control." The processes for the storage and retrieval of controlled items from library storage are also described in Section 4.11 "Media Control."

6.3.2 Configuration Change Control

All software and media related to a project are identified by a unique number.

Software configuration controls are put in place as soon as software development is initiated on a project. Configuration controls include:

1. Limiting access to master copies of media or documentation.

2. Placing duplicate (backup) copies of media in physically different locations to protect against hazards such as fire. Creating regular backups of work in process to minimize hazard loss or loss due to hardware failures.

3. Using software tools to detect and eliminate software viruses.

4. Maintaining a master list of software placed under configuration control for any given project, which is updated until the product is shipped (and a Computer Code Certificate and IV&V Report are issued).

5. Controlling the configuration of any support software or software tools used in the development, integration, testing, and documentation of the software system.

6. Control of previously developed software, purchased software, and NRC approved software is described in Reference 4.

7. The processing of requests for deviations and waivers from the provisions of specifications or supplier contracts is addressed by the Software Change Request Procedure and Reference 4, respectively.

Changes to a software item are controlled through the use of a Software Change Request (SCR) as follows:

SOFTWARE CHANGE REQUEST PROCEDURE

All changes to software performed after initial release will be performed in accordance with the following steps. These activities shall be performed via an automated process.

Step 1: Software Change Request Initiation

The requester of a change must complete EXHIBIT 6-1 SOFTWARE CHANGE REQUEST FORM, by providing the following information: (the exhibit represents the minimum information required.)

1. Name of person requesting the change
2. Date
3. Software system affected
4. Modules affected
5. Documents affected
6. Reason for the change

7. Description of the change
8. Classification of the change

SCRs may be initiated by an Exception Report, or by a request for enhancement.

Step 2: Analysis and Evaluation of a Change Request

The process for analyzing and evaluating a change request is defined in the Software Maintenance Plan, Section 9.3, "Analysis."

Step 3: Software Change Request Approval/Rejection

The SCR is routed to the CCB for approval/rejection of generic software. Project-specific software goes to the Lead SW engineer for approval/rejection.

The CCB determines the feasibility and appropriateness of the change for generic software, while the Lead SW engineer determines the feasibility and appropriateness of project-specific software changes. They sign the form for approval/rejection. Rejections must include an explanation for the rejection. The PM or Program Manager must approve Customer/User requests for changes. The other roles and responsibilities of the CCB can be found in subsection 6.2.2.12, "Configuration Control Board," of this SCMP.

Step 4: Software Change Implementation

After approval of the SCR, the EPM will schedule the change and the personnel responsible for implementing the change. After implementation, the changed software and accompanying documentation will be submitted for inclusion in controlled system files and documentation. The associated change request, and the names and versions of the affected items, will be documented in the SCR. The release date and the new version's identifier are found in the Software Release Records. The verification date is documented in the IV&V report, which will reference the Software Release Record.

Step 5: Revised System Baseline

The SCR forms will be used as the basis to track all software changes and to verify that changes have been properly implemented and that documentation has been updated.

6.3.3 Configuration Status Accounting

Information on the status of documentation and software configuration items is to be maintained by the IV&V group or design group ELM or designee. This may be accomplished for simple projects by

maintaining lists using commonly available word processing or spreadsheet programs or by Computer Aided Software Engineering (CASE) tools available on the development platforms. For larger projects, database programs may be utilized to simplify the maintenance process. In all cases, information on the status of documentation, software items, Exception Reports, Software Release Records and error notifications shall be made available for use in the Configuration Management Release Report. These reports when produced shall document the system status at any given time and be maintained by the IV&V Group or design group ELM, or designee, for inspection by the customer/user and any auditors.

6.3.4 Configuration Audits and Reviews

1. IV&V reviews shall be performed in accordance with this Software Program Manual IV&V procedures or a project specific IV&V plan.

2. Management and technical reviews shall be managed by the EPM in accordance with the Project Quality Plan (Reference 4) and this Software Program Manual.

3. External audits by customers or regulators shall be coordinated by QA or Licensing who will schedule personnel to be available if additional support is required.

4. In-process audits shall be performed by the Quality organization to verify the consistency of the design process and for proper implementation of the software QA process. Quality audits may be held at any time by the Quality organization to confirm that the software development guidelines, including configuration control, Independent Verification and Validation, and Software Quality Assurance are being adequately executed. These shall be documented in an audit report.

5. A functional review shall be performed in accordance with subsection 4.6.2.5 by the IV&V team prior to shipment to verify that all requirements specified in the Software Requirements Specification for the software configuration items have been met. This will be accomplished by the IV&V requirements traceability analysis.

6. Physical reviews shall be performed in accordance with subsection 4.6.2.6 to verify that the as-built software and its documentation are complete, meet all project technical requirements, and that the software change control process was adequately followed.

All audits and reviews shall be documented by meeting minutes or formal report, which will be tracked by the EPM for resolution of outstanding issues.

6.3.5 Interface Control

The EPM is responsible for coordination of communications and information transfer between the following entities to provide effective control of external interfaces to the Common Q™ System:

1. The project team and the customer
2. The project team and sub-vendors/subcontractors
3. Hardware, software, and functional engineering design personnel within the project team

Interface communications external to the design team shall be documented with numbered and dated correspondence. Correspondence logs are controlled via Reference 4. Interface between the design team and the independent IV&V team shall use either written correspondence or automated tools, e.g., Exception Report database.

The hardware configuration which supports the documented software configuration for a deliverable computer system must be controlled using drawing control procedures identified in Reference 4. The hardware configuration supporting software tools shall be documented in the user manual.

Interface communications external to NA shall be documented. Interface between the EPM and the independent IV&V team shall also use written correspondence.

The software requirements and design documents shall define the following for each external interface of the Common Q™ System:

1. Interface design
2. The organizations involved

The IV&V team ELM is responsible for configuration control of communication interface software for the Common Q™ System side of the interface. All documentation on the interface, that was generated external to NA, shall be placed in configuration control.

6.3.6 Subcontractor/Vendor Control

6.3.6.1 Subcontractor Software

New Protection class and Important-to-Safety class software to be developed by a subcontractor shall meet the requirements of Reference 4 and shall be maintained by the subcontractor prior to shipment to Westinghouse using an SCM plan judged by the IV&V team to be equivalent to this SCMP. Westinghouse does not need to plan for how proprietary items will be handled for security of information and the traceability of ownership because Westinghouse owns the rights of subcontracted software.

6.3.6.2 Vendor Software

Existing vendor software previously developed may be used "as-is" or modified prior to incorporation within the software system. This may include software that is supplied in support of the delivered system or may be integral to the delivered system, such as operating systems, compilers, database software, etc.

Existing vendor software, which is modified prior to delivery, must have a documented plan for modification. The plan must be evaluated and judged by the IV&V team to be equivalent to the SPM software change procedures.

All vendor software shall be evaluated to determine the adequacy of this software. The level of evaluation is determined by the following classifications:

- Development Tools (compiler, linker, loader, etc.) shall not require extensive IV&V or testing to qualify their use, since the end product is extensively tested and the tool is not used in on-line operation of the system.

- Software to be incorporated into the delivered product "as-is" or with modifications by design group is to be evaluated to determine the adequacy of this software for the intended application. This evaluation shall be performed in accordance with Reference 3. The evaluation is documented in a Commercial Grade Dedication Report.

6.3.7 Release Management and Delivery

The build, release, and delivery of software products will be formally controlled through Work Instructions. Master copies of code and documentation shall be maintained for the life of the software product using an approved software configuration management tool and the Electronic Data Management System (EDMS), respectively. The code and documentation that contain security-critical functions shall be handled, stored, packaged, and delivered according to Section 12, Secure Development and Operational Environment Plan.

6.4 SCM SCHEDULES

The project schedule shall include major SCM activities that depend on other activities in the project. SCM milestones that shall be indicated on the project schedule include:

- Establishment of a configuration baseline, and
- Implementation of change control procedures.

Establishment of the CCB is defined in the PQP.

The QA department controls configuration audit start/completion dates.

6.5 SCM RESOURCES

The IV&V team ELM shall identify the appropriate tools, techniques, and methodologies that may assist in SCM activities. These may include commercially available products for code control, version identification, and media backup/control. If project specific tools, techniques and methodologies are not identified, the following are to be used (minimum requirements):

1. At project baseline, a list of software shall be maintained by the IV&V team or design team ELM in the design file to include module name, version and revision, and executable file identification. In addition, a list of software tools (compilers, linkers, loaders, etc.) and their version/revision shall be maintained by the IV&V team ELM and kept in the design file. These lists may be maintained by commercially available word processing, spreadsheet, or database programs.

2. Software backups of all program files, including tools, shall be started upon system baseline and shall be updated on a regular basis, with changed files backed up on a weekly basis as a minimum. Backup methodology (saving all files or those which have changed in the last "x" days) shall be established by the EPM. Backup files shall be kept in a separate building from the development location. Backups may be kept as read-only files on a computer network as long as the file locations are physically separate from the software development location.

Documentation is to be maintained physically and electronically in accordance with Reference 4.

6.6 SCM PLAN MAINTENANCE

The Quality Assurance department is responsible for monitoring that Common Q™ software design groups are adhering to this plan. This plan shall be updated when nuclear and industry standards for software configuration management have been changed. The IV&V team ELM shall evaluate the new standards and determine if this plan requires revision. If a revision is required then this plan shall be revised and approved by both the IV&V team ELM and the Quality Assurance department. The revised plan shall be distributed to all Common Q™ EPMs doing software design work.

(Last Page of Section 6)

SECTION 7
SOFTWARE TEST PLAN

7.1 INTRODUCTION

7.1.1 OVERVIEW

This plan shall define the process for testing Common Q™ safety systems. This plan identifies testing activities and test documentation required to verify and validate a Common Q™ safety system throughout the software life cycle.

7.1.2 SCOPE

The scope of this plan includes testing processes for both Common Q™ platform components and applications developed with the Common Q™ platform. The information presented in this plan provides the prescribed details for a testing program.

Administrative software used for purposes such as ordering, scheduling, configuration management, and project management is not part of a delivered safety system and is, therefore, excluded from the testing requirements this plan imposes. Commercial applications software for use in software development, database management systems, word processing, and commercially purchased computer-aided design (CAD) systems – such as Microsoft® Excel, Word and AutoCAD® software – are also excluded.

7.1.3 OBJECTIVE

The Common Q™ safety systems testing process validates the functional requirements of the Common Q™ safety systems applied to a specific project and/or a component being developed for the Common Q™ platform. This plan is intended to guide a qualified test team to prepare detailed test procedures that conform to the Common Q™ safety systems criteria.

Project-specific testing requirements shall be included in a project-specific Test Plan.

7.2 TESTING PROCESS OVERVIEW

7.2.1 Organization

Organization for the Common Q™ testing process is per Section 2; whereby the IV&V team is responsible for testing activities.

Subsection 5.4.1 provides details of the organizational structure and interfaces between the design, verification, and testing processes.

7.2.2 Staffing and Training

The IV&V Test Team is made up of members assigned to the IV&V team to perform testing functions (preparation of plans, procedures, and reports; conducting tests). Additional duties and qualifications shall be based on project-specific requirements.

7.2.2.1 Duties

One or more people assigned to the IV&V Test Team shall fulfill the following organization functions: IV&V Lead Test Engineer and IV&V Test Engineer.

Engineering staff assignments to the IV&V Test Team shall be based on the technical field of experience and current work assignments.

7.2.2.2 Qualifications

IV&V Test Team members shall receive any required project-specific training. All training shall be documented, and the training records shall be maintained.

Designated IV&V Test Team members shall have training on the software testing tools that may be used during the testing process. Designated IV&V Test Team members shall require specialized training in the requirements traceability process for tracing requirements to test case preparation, and test case reporting.

7.2.3 Responsibilities

The IV&V team manager shall track the overall status of the IV&V test effort. The IV&V team leader shall inform the EPM of IV&V status and request documented resolution of IV&V issues. The IV&V team leader shall communicate guidance and issue resolution to the IV&V Test Team. The IV&V team leader shall determine IV&V Test Team member task assignments, and participate in preparing and maintaining the testing elements of the project schedule.

The IV&V team leader is responsible for identifying the proper qualifications of the IV&V Test Team members.

The ELM shall provide the environmental needs identified in subsection 7.2.5 to the IV&V Test Team.

7.2.4 Schedule

A detailed test schedule prepared by the IV&V team leader shall be available for the project team to integrate into the project schedule. The IV&V team leader and project team shall be actively maintain and update the test schedule. The IV&V Lead Test Engineer shall be involved with any decision that causes a deviation to testing the order described below.

Testing activities begin with preparing test procedures for modules that are developed for a Common Q™ safety system. Formal module validation testing for protection class software begins with the design team's release of the software module.

The following outlines the prescribed testing sequence for Common Q™ safety systems (see subsection 7.3.1 for a description of each testing level.):

- Module Test – A module test is completed before the software module is used in an application released for validation testing. If not, then impacts shall be documented and incorporated in a regression analysis on downstream validation testing.

- Unit Test –Unit Testing is completed before the Integration Test is completed. If not, impacts on unit changes shall be documented and incorporated in a regression analysis on completed testing and downstream validation testing.

- Integration Test – The Integration Test is executed before running the system validation test or FAT. If not, then impacts shall be documented and incorporated in a regression analysis on downstream validation testing.

- System Validation Test – The System Validation Test shall be completed before SAT is completed.

- Factory Acceptance Test (FAT) – The FAT is to be executed on a deliverable system and must be completed and meet its approved requirements before the customer accepts the system. The FAT is typically performed in the factory but some portion of the test can be performed at site if agreed to with the customer. When performed on a deliverable system, the System Validation Test can fulfill the role of the Factory Acceptance Test.

- Site Acceptance Test (SAT) – The SAT shall be executed when installation of the safety system at the customer's site is complete.

Depending on the system's size, the Unit Test, Integration Test, System Validation and FAT can be consolidated as defined in project-specific test plan.

7.2.5 Testing Environment

This section describes the properties of the testing environment that shall be addressed in the test procedures or test plan. Each procedure or plan shall identify

- The physical characteristics of:
  - The specific testing hardware
  - The communications
  - The system software
  - Any other software or supplies needed to support the test
- Special testing needs, such as:
  - Test tools
  - Software
  - Publications
  - Documentation
  - Testing area
- The hardware or software configuration (or both) undergoing testing shall be identified in the individual test procedures, test plan, or equivalent test configuration control document (Test Configuration Record) in sufficient detail to completely capture the configuration that is being tested.

7.2.5.1 Testing Hardware

Each test procedure shall specify the hardware requirements for conducting the test. The following guidelines shall be used for the various testing levels (see subsection 7.3.1):

- Module Tests – A software module test shall be conducted on a test bed configured with the appropriate software test tools that provide structural test (code coverage) results. A software module shall undergo functional testing – providing input test signals and recording output values. These tests are included in the description of Component tests in IEEE Std. 1012 (Reference 8).
- Unit Tests – These tests shall be conducted on a test bed equipped with an AC160 processor that is connected to an I/O Simulator providing input test signals and recording output values. These tests are the equivalent of the description of Component tests in IEEE Std. 1012 (Reference 8).

- Integration Tests – These tests shall be conducted once as design validation on deliverable hardware or functionally equivalent hardware, assembled as a cabinet or single-channel. These tests are the equivalent of the description of integration tests in IEEE Std. 1012 (Reference 8).

- System Validation Tests – These tests shall be conducted once as design validation on deliverable hardware or functionally equivalent hardware configuration assembled in cabinet(s) and configured with the application software for the purpose of certifying a design. These tests are the equivalent of the description of System tests in IEEE Std. 1012 (Reference 8).

- Factory Acceptance Tests – These tests shall be conducted on the deliverable hardware assembled in cabinet(s) and configured with the application software. The integration and system validation test can be credited for applicable parts of the Factory Acceptance Test (FAT) when conducted on deliverable hardware. FAT is the equivalent of the description of Acceptance tests in IEEE Std. 1012 (Reference 8).

7.2.5.2 Security

Section 12 provides the Secure Development and Operational Environment program for Common Q™ Systems.

7.2.6 Test Tools

Test equipment that a test procedure specifies for use and which requires calibration shall be calibrated and maintained under configuration control throughout the testing process.

7.2.7 Features and Functions to be Tested

All testable requirements for Common Q™ safety system features and functions shall be tested with explicit acceptance criterion. Subsection 7.3.1 provides details on requirements testing. The requirements shall be derived from the requirements traceability process. Each testable feature and function identified within the Requirements Traceability Matrix (RTM) shall be tested with a procedure that is traceable to the item within the RTM. Maintaining the RTM shall provide evidence of complete test coverage of Common Q™ safety system features and functions.

7.2.8 Risks and Contingencies

Regression analysis shall be performed to determine extent of retesting activities that may be necessary to re-verify and/or re-validate any changes to a tested element. Design modifications, or detection of latent design errors or programming bugs may have been brought about these changes.

7.2.9 Standards, Practices, and Conventions

Testing effort standards, practices, and conventions that differ from those stated in this process shall be specifically stated and justified in a Project Quality Plan. These differences shall be summarized in the IV&V summary report.

7.3 TESTING PROCESS ACTIVITIES AND TASKS

Testing can be divided into two categories: functional testing and structural testing.

Functional Testing (black box testing) shall be used to determine that a module or system has functional performance consistent with the requirements specified for the module or system. Test cases for functional testing shall be derived from the requirement specifications and shall be based on manipulating test inputs and monitoring test outputs.

Structural Testing (white box testing) shall evaluate the internal structure of a code module and is only used for module tests. Structural testing shall provide one hundred percent of branch execution within the code module.

7.3.1 Testing Methodology

The testing methodology shall follow a low-level to high-level scheme, from component up through system validation and FAT testing, as shown in Table 7.3-1. Since some safety system designs involve functional redundancy, a redundant code module shall be analyzed for differences from the tested code module. When differences are apparent, the documented analysis shall identify additional testing procedures.

Table 7.3-1. Testing Levels

Test Type	Level 1	Level 2	Level 3	Level 4
Software Component	Module Test	Unit Test	---	---

Table 7.3-1. Testing Levels

Test Type	Level 1	Level 2	Level 3	Level 4
Integration	---	---	Integration Test	System Validation Testing
Manufacturing	---	---	Hardware Tests and FAT	

Formal testing shall begin when all associated system hardware, software, and documentation is placed under configuration control and released for testing.

Modification of the test items or test environment (comprising hardware, software, and/or test procedures made during the testing process) shall be performed according to the appropriate change control procedures described in the SCMP.

7.3.1.1 Module Test

A module test shall address the requirements specified in the software module document.

A module test shall combine functional and structural testing. Functional and structural testing shall be accomplished using test cases with varying input values that exercise the software module's boundaries and internal branches and paths.

The following test items shall be included in a module test:

- Initialization – all variables, pointers, and I/O points shall be initialized
- Range Checking – all inputs shall check for maximum and minimum values
- Error Handling – potential errors (such as divide-by–zero or out-of-range) shall be handled with known consequences
- Calculations – the accuracy of any calculation performed shall be verified
- Timing – a module's timing requirements shall be verified
- Branch Coverage

7.3.1.2 Unit Test

A Unit Test shall address the safety system requirements documented in the Software Requirements Specification.

A Unit Test is a functional test that verifies the application program's functionality.

The following test items shall be included in a Unit Test (if applicable):

- Supervisory Logic – supervisory logic implemented in an application program shall be tested as applicable for completeness and correctness
- Process Logic – process logic implemented in an application program shall be tested as applicable for completeness and correctness
- Quality Signals – quality signals created in an application program shall be tested as applicable for completeness and correctness
- In-Test Signals – in-test signals created in an application program shall be tested as applicable for completeness and correctness

A unit software code review shall be conducted. This review shall trace the software functionality to the software design, software requirements or the functional specifications. The code review shall verify that the application program only consists of software modules that are qualified for use as part of the Common Q™ safety system. The code review process shall provide reasonable assurance that no unintended functions exist within the application program. The Code Review shall also be credited for those safety system software requirements that were determined to be validated through inspection, as opposed to testing, by the IV&V Test Engineer.

7.3.1.3 Integration Test

An integration test is a functional test that verifies the released software's integration with the production hardware or with a system that is functionally equivalent. A functionally equivalent system can be a test bed or an equivalent set of production hardware (e.g. a unit of the same design for a different site deliverable system). A test bed shall be configured with hardware that provides functionally equivalent configuration to the production hardware for the testing performed.

An integration test shall address the safety system requirements documented in the System Requirements Specification.

Integration testing is used as part of system validation testing when validating the design and as part of the FAT testing to demonstrate the deliverable system has been properly integrated.

The Integration test can be segregated into tests that are performed on a cabinet level, on a division or channel level, or on a system level. For tests on a channel level, cabinets within a safety system division shall be interconnected and integrated for this test. Functions implemented in a single cabinet within a division or across multiple cabinets within a division shall be tested. Communications between cabinets within a division, data flow, control functions, signal loops, redundancy, interdivisional voting logic, and fault tolerance shall be tested. Functions implemented across multiple divisions shall be tested with the system fully integrated during the system validation test or FAT. Functions shall be tested by confirming the correct relationship between test input and output signals. Each input signal shall be exercised to verify mapping with expected outputs.

7.3.1.4 System Validation Test

The system validation test is a set of tests developed to validate the hardware design, software design, and the system integration at the functional level. The system validation test shall address the safety system requirements documented in the System Requirements Specification.

Aspects of system validation testing can be performed on a single division to show compliance of functions that are contained within one division. The system validation testing is also performed on multiple divisions to show compliance of functions that require communication with other divisions.

The system validation test shall test the integration of the cabinets in the safety system as defined by the project-specific test plan.

The system validation test shall verify that the cabinets in the safety system divisions (as defined the project-specific test plan) satisfy system-level functional and performance requirements. The test shall verify correct communications between cabinets in different divisions.

System validation functional testing shall focus on system-level functional requirements requiring cabinet interaction both within the division and across divisions.

Testing shall verify system boundaries to other I&C systems, communications between divisions (including interface loading), data flow, control functions, signal loops, redundancy, interdivisional voting logic, and fault tolerance incorporated in the system's design. Overall system time response shall be verified.

The following test items shall be included in the system validation test:

- Safety Functions
- Communications
- Displays
- Diagnostics
- Performance
- Error Handling – potential errors shall be handled with known consequences
- Communications – all defined outputs shall be broadcast and received correctly within the channel
- Redundancy – all shared inputs shall produce the same output from redundant processors
- Diversity – all functionally diverse signals shall be verified for correctness in termination

See EXHIBIT 7-1 COMPARISON OF SYSTEM VALIDATION TEST AND FAT for a detailed description of the tests performed during system validation testing and FAT.

For system validation test to be credited as FAT, it must be performed on the delivered equipment.

As an alternative to functional testing with production hardware, a system validation test can be performed with a test bed. This test bed shall be a functionally equivalent configuration to the production hardware. Alternatively, system validation testing can be performed on any of the first of a kind deliverable system. As design changes are introduced, regression analysis needs to be performed to determine what tests need to be repeated or introduced to maintain the level of system validation achieved during the first of a kind test program. The system validation tests required by the regression analysis may be performed on the deliverable equipment as a separate section of the FAT or on surrogate equipment consistent with the regression testing methods described in subsection 7.3.2.2.

7.3.1.5 Factory Acceptance Test (FAT)

The purpose of the FAT is to demonstrate that the complete system is integrated and functional. To this end, the optimum scenario is to perform this test in the manufacturing facility. Prior to acceptance of equipment by the customer, a Factory Acceptance Test (FAT) is performed as a manufacturing test to provide evidence to the customer that the system meets its requirements and provides confidence that the site installation and integration activities will be successful. FAT includes tests that are performed on the deliverable system for each deliverable system. The FAT test, together with the documentation of the prior V&V activities (module tests, unit tests, software code reviews, integration testing, and system validation testing, etc.) demonstrate full compliance to the requirements. Upon agreement of the customer, some or all of the FAT activities may be deferred to site following installation.

FAT is performed to:

- Demonstrate that the system has been manufactured correctly and is acceptable to the customer

- Demonstrate (in conjunction with V&V) compliance to requirements for customer acceptance
- Reduce the risk associated with deferring compliance demonstration to the site activities (e.g., SAT, preoperational testing, etc.)
- Demonstrate aspects of the design that would not be practical once full integration is achieved due to limitations on interfaces that are connected in the plant.

The completeness of the FAT is demonstrated by a combination of the tests performed and reference to prior tests on the first application system that remain valid because the design is identical in all relevant aspects. Such references must be specific as to procedures and test cases or a reference trail. The results of these reference tests must be kept under configuration management, and any open items arising from the test must be either resolved or carried forward to the follow-on system.

The following test items shall be included or demonstrated in the FAT:

- Safety Functions
- Communications
- Operability of Displays
- Diagnostics associated with hardware specific inputs (door alarms, temperature alarms, breaker status, etc.)
- Performance (accuracy, time response, etc.)

See EXHIBIT 7-1 COMPARISON OF SYSTEM VALIDATION TEST AND FAT for a detailed description of the tests performed during system validation testing and FAT.

7.3.1.6 Site Acceptance Test (SAT)

The SAT is a two-part test verifying correct functionality and performance after the system is installed at the customer's site. The site test personnel shall define and control the test. The primary intent of this test shall be to validate that the equipment was not damaged during shipment or installation. External system interface testing shall be specified in the SAT procedure.

7.3.2 Pass/Fail Criteria and Regression Testing

7.3.2.1 Pass/Fail Criteria

The safety system must satisfy specified functional and performance requirements, (such as those identified in the project's System Requirements Specification). Specific pass/fail criteria shall be provided in the applicable test procedure. For expected numerical test results, an acceptable range shall be provided. For expected test results that are logical conditions or alarm states, the specific digital value or state shall be provided.

Pass/fail acceptance criteria shall be captured in the test procedure's data sheets.

If a pass/fail criterion is not met during a test, the failure shall be clearly captured in the Test Log and Test Report, and entered in the Anomaly Reporting system for tracking purposes.

7.3.2.2 Regression Testing

Safety System changes can occur for several reasons. For example, changes can be made at the Customer's direction or as a result of problems discovered during testing. It is normal for hardware and software modifications to be required during the system test period. All changes shall be formally documented and controlled according to the SCMP and the SMP.

Any time a problem is found and corrected or a change is made in the system, a regression analysis is performed and documented in the Exception Report (ER). Once it is determined what subsystems and elements are affected, a review of the appropriate test procedure shall be performed to determine the changes in testing.

If the scope of the regression validation is at the unit level, then code inspection of the differences can be an acceptable method.

Original tests are performed on deliverable or surrogate hardware, as defined in the safety system test procedures. The deliverable hardware may not be available once the original tests have been completed. In this case, regression testing on surrogate equipment is permitted to be performed. Surrogate equipment performance and interface loading must be equivalent to the deliverable equipment for the level of testing performed.

(Last Page of Section 7)

SECTION 8
SOFTWARE INSTALLATION PLAN

8.1 PURPOSE

The purpose of this plan is to describe the installation of software for the Common Q™ system.

8.2 OVERVIEW

This plan covers:

- Loading operating system software into AC160 processor modules.
- Loading application program software into AC160 processor modules.
- Loading operating system and application program software into the Flat Panel Display Systems (FPDS).

8.3 AC160 SOFTWARE INSTALLATION

8.3.1 AC160 Base Software Installation

[

]a,c

8.3.1.1 Loading the AC160 Communication System Software (CS)

[

]a,c

8.3.1.2 Loading the AC160 Base Software (PS)

The operating system software is loaded with the image file documented in the Software Release Letter using approved load instructions. [ ]a,c

[

]a,c

8.3.1.3 Loading the AC160 Software Library Options (PS)

[

]a,c

8.3.2 AC160 Application Software Installation

The application program is installed in each PM after the PS operating system software and the library options are loaded.

8.3.2.1 Installation of AC160 Application Software

The Function Charter Builder (FCB) is used to load the application program using approved load instructions. [

]a,c

The Application Program is started using the approved instructions for starting an application.

8.4 FLAT PANEL DISPLAY SYSTEM (FPDS) SOFTWARE INSTALLATION

8.4.1 FPDS Operating System Software Installation

[

]a,c

8.4.2 Loading the FPDS Application Software

[

]a,c

[

]a,c

(Last Page of Section 8)

SECTION 9
SOFTWARE MAINTENANCE PLAN

9.1 INTRODUCTION

The Software Maintenance Plan specifies the requirements for the maintenance and use of Protection class and Important-to-Safety class software used in Common Q™ Systems.

Normally, the ELM is responsible for Common Q™ software during the Operation and Maintenance Phase. However, for extensive software modifications an EPM may be assigned. Therefore, any activity that is designated as an ELM responsibility may be assigned to an EPM.

Exception Reports shall be prepared to document all software anomalies discovered during the Software Operation and Maintenance Phase. Anomalies may include test deviations, system malfunctions, or inconsistencies between the software and documentation. If a software change is required to resolve the exception report, then the Software Change Request is issued. Software Change Requests are required to initiate any software change after the initial software baseline is established.

9.2 PROBLEM/MODIFICATION IDENTIFICATION, CLASSIFICATION AND PRIORITIZATION

A four-level priority scale shall be used in the classification of software problems (refer to EXHIBIT 6-1 SOFTWARE CHANGE REQUEST FORM). Metrics and measures for this phase are specified in section 4.5.2.4.

9.2.1 Input

Input for the problem/modification and classification phase shall be a Software Change Request (SCR). A description of the SCR process is found in subsection 6.3.2.

9.2.2 Process

The SCR shall specify:

1. An identification (SCR) number
2. A classification number identifying the maintenance type and prioritization
3. A description of the software modification that describes the magnitude of the change.

The SCR is submitted to the CCB for approval of generic software, while project-specific software is submitted to the Lead SW engineer for approval. They can accept/reject the SCR or request further clarification. If the SCR is approved, then the modification is scheduled by the EPM.

9.2.3 Control

An SCR log shall be maintained for the specific Common Q™ system implementation. The Platform Lead shall confirm that the approved SCR is entered into this log for any internal generic software changes. The Lead Software Engineer shall confirm that the approved SCR is entered into the SCR log for any project-specific software changes.

9.2.4 Output

The approved SCR is the output to this process. The original exception report shall be attached to the SCR if applicable. The EPM should be provided an estimate for the modification as input into the next phase.

9.3 ANALYSIS

This phase of Software Operation and Maintenance involves a feasibility and detailed analysis of the modification. If the modification is a correction to an error and the requirements remain the same, this phase of software maintenance may not be applicable.

9.3.1 Analysis Input

Input to the analysis phase of the maintenance process shall include:

1. Approved SCR
2. Entry of the SCR into the SCR log
3. Any relevant project or system documentation

9.3.2 Analysis Process

This section specifies the process requirements for analyzing the modification.

9.3.2.1 Feasibility Analysis

If the scope of the modification requires extensive software changes, a Project Quality Plan (Reference 4) shall be developed; otherwise, the SCR “summary of requested change” shall suffice. It may also be

possible to use an existing Project Quality Plan previously published for the project. In addition to the required information, the Project Quality Plan should address the following if applicable:

1. Impact of the modification
2. Alternate solutions
3. Analysis of conversion requirements
4. Safety and security implications
5. Human factors
6. Costs
7. Value of the benefit of making the modification
8. How the design, implementation, testing and delivery of the modification is to be accomplished with minimal impact to current users.

9.3.2.2 Detailed Analysis

If the modification is a change to existing requirements, then firm requirements for the modification are defined in revised System and/or Software Requirements Specifications. The SRS shall identify the software elements that require modification. Any safety and security requirements shall be included in these documents.

During this phase a test plan may need to be developed in accordance with subsection 4.3.2.2 that specifies the test strategy for the modification including any regression testing requirements. For protection class software, the test plan shall address any requirements for module testing. If the change is limited to error corrections, then a regression test plan can be specified in the Error Report.

If necessary the Project Quality Plan shall be updated to reflect any changes to the planned implementation (design, implementation, testing and delivery) of the modification such that current users are minimally impacted (see subsection 9.3.2.1).

9.3.3 Analysis Control

At this phase of the analysis, the IV&V team shall review any changes to the requirements specifications and review the test plan(s) as defined in subsections 5.5.4 and 5.5.8.

The relevant version of project and system documentation from the appropriate configuration control organization (NA or customer) shall be retrieved (refer to Section 6 for Software Configuration Management). The design Team shall review the proposed changes and newly revised requirements specifications. The design Team shall then consider the integration of the proposed change within the existing software.

The Project Quality Plan shall be reviewed by the EPM for any changes to the risk analysis after the Design Team reviews the proposed changes and revised requirements. If the change is limited to error corrections, then a review of the software release record can suffice.

9.3.4 Analysis Output

The output of the analysis phase of software maintenance includes the following documents if the modification is the result of a change in requirements.

1. Project Quality Plan
2. Revised System and/or Software Requirements Specifications
3. Test Plan
4. IV&V Requirements Phase Report including RTM

9.4 DESIGN

This section defines the design requirements for software maintenance. Metrics for this phase are defined in subsection 4.5.2.4. If the modification does not affect the design of the software, then this phase of software maintenance may not be applicable.

9.4.1 Design Input

All outputs from the identification and analysis phases are used as inputs into this phase of software maintenance.

9.4.2 Design Process

At this phase the affected software modules are identified and the SDD is revised to incorporate the modification into the design.

For protection class software, module test procedures are created/modified in accordance with the test plan and Reference 21. Unit and integration test procedures (with test cases) are developed in accordance with Section 5.8 to test the modification in accordance with the test plan.

At this phase, the Design Team shall identify any installation or user documentation that must be revised to incorporate the modification.

9.4.3 Design Control

The IV&V team shall review the revised SDD as defined in subsections 5.5.4 and 5.5.8 and the test procedures for the modification as defined in subsections 5.5.6 and 5.5.8.

9.4.4 Design Output

The output of the design phase of software maintenance shall include:

1. Revised SDD
2. Test Procedures
3. Design Phase IV&V Report including Requirements Traceability Matrix

9.5 IMPLEMENTATION

This section defines the requirements for the implementation phase of software maintenance. Metrics for this phase are defined in subsection 4.5.2.4.

9.5.1 Implementation Input

The inputs to the implementation phase shall include all outputs from the identification, analysis and design phases (if applicable).

9.5.2 Implementation Process

The implementation phase shall include the following sub processes.

9.5.2.1 Coding and Module Testing

At this phase the source code is modified and compiled, and new executables generated. For protection class software, module test procedures are run and results documented. For other software classes, informal module testing may be conducted. The IV&V activities related to module testing for protection class software is performed in accordance with subsections 5.5.6 and 5.5.8.

9.5.2.2 Integration

Integration is the process of running the revised software in an integrated system environment. It includes informal integration and regression testing to validate that the system as a whole is fully operational prior to system testing. Any anomalies shall be documented using the Exception Report form and changes shall conform to the software configuration management plan in Section 6.

9.5.2.3 Documentation

Any user, training or installation documentation that is impacted by the modification shall be revised at this time. It shall be submitted to the IV&V team for review per subsection 5.5.5.

9.5.2.4 Risk Analysis and Test-Readiness Review

The EPM shall review the status of the integration and determine when the software is ready for official system testing. In addition, the Project Quality Plan shall be updated if the risk assessment has changed.

9.5.3 Implementation Control

The IV&V activities associated with the implementation phase of the software life cycle as defined in subsections 5.5.5 and 5.5.8 shall be performed to verify implementation control. The IV&V team ELM shall be responsible for all software being under software configuration management control in accordance with Section 6.

9.5.4 Implementation Output

The outputs of the implementation phase of software maintenance shall include:

1. Updated software
2. Updated module test procedures (if required)
3. Updated user, training, and installation documentation (if required)
4. Implementation Phase IV&V report

9.6 TEST

At this phase, formal testing is performed on the new software system.

9.6.1 Test Input

All outputs from the previous phases are used as inputs into this phase of software maintenance.

9.6.2 Test Process

During this phase the IV&V team revises or develops new validation test procedures with test cases (if required) to test the modification in accordance with Section 5.8.

After the test procedures have been released, the validation tests are performed on the new software system according to the test plan. Any test exceptions shall be documented using the Exception Report form and changes shall conform to the software configuration management plan in Section 6.

After the completion of the validation test, a test report shall be issued and reviewed in accordance with subsection 5.5.6.

9.6.3 Test Control

Validation tests shall be conducted by the IV&V team for protection and important to safety software. Any test exceptions shall be documented using the information required by Exception Report form (EXHIBIT 11-1 EXCEPTION REPORT) and changes shall conform to the software configuration management plan in Section 6. The test report shall be issued and reviewed in accordance with subsection 5.5.6.

9.6.4 Test Output

The outputs for the validation test phase of software maintenance are the same as the test phase IV&V outputs specified in subsection 5.5.6.

9.7 DELIVERY

This phase of software maintenance is the final acceptance of the modification prior to shipment to the customer. All metrics have been collected in accordance with subsection 4.5.2.4.

9.7.1 Input

The inputs to this phase of software maintenance include the outputs from all previous phases.

9.7.2 Process

Physical reviews on the new software system shall be performed according to subsection 4.6.2.6. The users of the software shall be notified in accordance with Section 11. An archival version of the software shall be performed in accordance with Section 6.

9.7.3 Control

In addition to the physical reviews, the IV&V team shall perform the activities associated with the Installation and Checkout Phase, subsection 5.5.7.

9.7.4 Output

In addition to the modified software, the outputs for the delivery phase of software maintenance include a final IV&V report and Code Certificate.

(Last Page of Section 9)

SECTION 10
DOCUMENTATION

10.1 GENERAL REQUIREMENTS

Software documentation shall be provided for all computer software to be used or delivered for Common Q™ systems. The author of a software document is responsible for updating a requirements traceability matrix (RTM), as described in subsection 5.4.5.3. The author's signature on a document shall signify that the RTM has been updated to reflect the design information contained in the document. All documentation shall comply with Reference 4.

10.2 SYSTEM REQUIREMENTS DOCUMENTATION

For a Common Q™ system the System Requirements are composed of Functional Requirements and Software Requirements. The Software Requirements may be included with the Functional Requirements as part of the System Requirements Specification (SysRS) or documented separately in the Software Requirements Specification (SRS).

Each requirement in the System Requirements Documentation shall be defined such that its achievement is capable of being verified by the SVVP.

10.2.1 System Requirements Specification (SysRS)

The System Requirements include:

- System Operational Requirements
- System Performance Requirements
- System Safety Requirements
- System Design Basis
- System Design Constraints

The System Requirements define high level system requirements Identifying those functions that will be performed by software and specifying the software safety critical actions that are required to prevent the system from entering a hazardous state, or move the system from a hazardous state to a non-hazardous state, or to mitigate the consequences of an accident.

10.2.2 Software Requirements Specification (SRS)

The Software Requirements Specification complies in content, but not format to Reference 6 and Reference 22. The SRS also complies with the requirements specified in the System Requirements Specification. The SRS is used as the source document for design of the software, including:

1. Description of major software components which reflect the software requirements
2. Technical description of the software (i.e. control flow, data flow, control logic, data structures)
3. Description of all interfaces and allowable ranges of inputs and outputs
4. Any other design items which must be translated into code
5. A description of the intended platform and programming language(s) expected to be utilized
6. Data necessary for final implementation such as setpoints
7. Abnormal conditions to be accommodated by the software shall be described, including resulting functional operations.
8. Plant input signal transient conditions to be accommodated by this software shall be described.
9. Software safety requirements that address System Safety Requirements.

10.3 SOFTWARE DESIGN DESCRIPTION (SDD)

The software design descriptions comply with the requirements of Reference 7. The SDD also complies with the System Requirements Specification and the Software Requirements Specification.

The purpose of the SDD is to depict how the software will be structured to satisfy the requirements of the SRS, including software safety requirements. The design shall be described such that it can be translated into software code.

The SDD is a detailed description of the software to be coded. It describes decomposition of the software into entities. Each entity is described by its type, purpose or function, subordinate entities, dependencies, interfaces, resources, processing and data.

Each design feature shall be described and defined, and each software safety design element identified that satisfy the software safety requirements, such that its achievement is capable of being verified and

validated per the SVVP. The adequacy of the SDD shall be verified against how the requirements of the software (documented in the SRS) are to be implemented in code, and how the design is traceable to the requirements in the SRS.

10.4 SOURCE CODE DOCUMENTATION

Source code documentation shall include software release records and code review reports.

Source code shall be traceable to the software design documented in the SDD and the requirements in the SRS. It shall include sufficient comments to provide the user of the source code with an understanding of the functioning and programming of each module. All source code, whether developed or modified from existing software, shall be documented in accordance with the coding standards listed in subsection 4.5.2.1.

10.5 SOFTWARE VERIFICATION AND VALIDATION DOCUMENTATION

Software IV&V documentation shall include Software IV&V Reports (SVVR), prepared according to Reference 8 as augmented by Reference 18.

10.5.1 Software Verification and Validation Plan

The Project Quality Plan (PQP) or a project specific SVVP shall identify the software items to be evaluated. The SVVP, Section 5, describes the IV&V evaluation and reporting activities. Verification review requirements and guidelines are described in Section 4.6 and Section 5. Validation tests to be performed shall be described in a separate Test Plan that is subordinate to the SVVP, and is included as part of the software IV&V documentation.

For custom software to be developed, the project specifics for IV&V shall be documented in the PQP or a project specific SVVP. If a project specific SVVP is written, then it must be referenced in the PQP.

For existing software to be modified, the PQP includes methods for verifying and validating modifications to this existing software.

The PQP shall provide adequate planning for the following, referencing Section 5 as appropriate:

- Software IV&V process for the various software categories described in subsection 4.1.1
- Software IV&V process for existing software to be modified and to be used “as-is.”
- Software IV&V process for prototype software

The PQP shall also define the tracking and recording process for the hardware configuration pertinent to the software verification and validation process during all phases of the software life cycle.

10.5.2 Software Verification and Validation Report

IV&V phase summary reports shall be issued by the IV&V Team throughout the software life cycle to document all IV&V activities. It shall summarize all validation test results, exception reports and corrective actions, verification review results, and the results of all quality audits (subsection 4.6.2.7). These reports shall form the basis for the development of a final SVVR upon installation and checkout life cycle phase.

The final SVVR shall be developed by the IV&V team in accordance with subsection 5.5.7.

10.6 USER DOCUMENTATION

User documentation is prepared according to Reference 9. The purpose of User Documentation is to provide sufficient information about the software to permit users to employ the code as it was intended. It shall be written by the design team. User documentation will be developed to the extent practical during the Test Phase and delivered to the user during the Installation and Checkout phase.

User documentation shall reference vendor documents and documents prepared as part of the project. Project prepared user documents shall be as follows. These documents can be combined into a single Technical Manual.

- User's Manual
- Installation and Operations Manual
- Maintenance Manual

User Documentation shall include all error messages and identify the necessary corrective-action procedures. Also, it shall provide the means for the user to report problems to Nuclear Automation.

If the end user will be maintaining the software, then the user documentation shall also include the System Build Specifications. The System Build Specifications provide the exact steps taken to build the program. This includes the names of modules and files, names of libraries, and scripts used to build the program.

10.7 SOFTWARE CONFIGURATION MANAGEMENT DOCUMENTATION

Project-specific SCMP details, such as the identification of specific SCM tools, shall be defined in the Project Quality Plan (PQP) or project specific SCMP. If a project specific SCMP is written, then it must be referenced in the PQP.

10.8 TEST DOCUMENTATION

This section describes the requirements for test plans and test procedures.

10.8.1 Test Plans

The requirements for test plans can be found in subsection 4.3.2.2.

10.8.2 Test Procedures

The requirements for Common Q™ module, unit, integration, system validation, and FAT test procedures can be found in Section 5.8.

10.9 SOFTWARE/DATABASE RELEASE RECORDS

Software Release Records are issued to document the software's configuration identity. The Software Release Record identifies:

- The software module or applicable code revisions
- The revisions of the applicable design documents
- The revisions of the tools that were used to create the software

The Database Release Records (DRR) are issued to document the installation configuration tables' configuration identity. These tables indicate I/O channel numbers, sensor and actuator connections and names, and other installation-specific configuration data.

10.10 COMPUTER CODE CERTIFICATE

Computer Code Certificates (see EXHIBIT 10-1 COMPUTER CODE CERTIFICATE for content requirements) are issued for Protection and Important-to-Safety software only. It shall identify the software classification of each software component listed on the certificate.

The issuance of a Computer Code Certificate allows the release of a configuration item for use in its intended application.

Software intended for limited use, such as in a single design analysis, may be used provided that the results as well as methods and/or formulas are documented in the design analysis in sufficient detail to allow independent verification. A Computer Code Certificate shall not be issued for such software on this basis alone.

(Last Page of Section 10)

SECTION 11
PROBLEM REPORTING AND CORRECTIVE ACTION

11.1 INTRODUCTION

There are two modes of exception reporting. The first is during the software development phase when validation testing is being performed and test exceptions are found. Section 11.2 describes the reporting process for these errors.

The second mode of error reporting occurs when a user discovers an error after software is approved for use. Section 11.3 describes this reporting process.

Errors shall be documented by completing a form consistent with the information required by EXHIBIT 11-1 EXCEPTION REPORT. The exhibit represents the minimum information required; the exact format of Exhibit 11-1 does not need to be followed provided all of the required information is present. The exception reporting procedure shall be implemented via an automated process.

11.2 ERROR REPORTING BEFORE SOFTWARE APPROVAL FOR USE

Discrepancies, deficiencies, or comments identified as a result of testing, review, or other means shall be documented in a formal manner. This includes any general discrepancies found outside of the normal IV&V test process. The following table illustrates the type of report required by each method:

Table 11.2-1. Error Reporting Methods

Method	Report
Verification Reviews	EXHIBIT 11-1 EXCEPTION REPORT
Validation Tests and FAT	EXHIBIT 11-1 EXCEPTION REPORT
General Findings	EXHIBIT 11-1 EXCEPTION REPORT

The appropriate configuration identification data (see subsection 6.3.1) for each deficient software item or document shall be included on the appropriate form (or report). The form (or report) shall also include a description of the observed deficiency, the name of the individual reporting the deficiency, and the date of the report finding.

In the case of an Exception Report, each form shall include space for a description of the resolution and any retest or review required after the resolution. If retest is performed, a copy of the test procedure or test case used shall be attached or referenced in the completed Exception Report. The steps taken to

cause the discrepancy to occur should also be included on the Exception Report form in order to reproduce the problem. These steps should be noted as best as possible if the problem is not repeatable.

The extent of the retest shall be determined by the appropriate team, either the design or IV&V team, based on the relative impact of the software change on the overall system operation. For Protection and Important-To-Safety software, all changes require complete system retest, unless otherwise justified in writing including steps to validate that new errors were not introduced.

A distinction is made between the Exception Reports filed by the IV&V team and those filed by others based on the verification status of the affected software. Software still under development and not yet released to IV&V is the responsibility of the design team. Exception reports filed by the design team for software not yet released to IV&V will be tracked and controlled by the design team.

11.3 ERROR REPORTING AFTER SOFTWARE APPROVAL FOR USE

Software errors may be found either internally or externally after the software Code Certificate has been issued. Errors found externally, i.e., by a customer, may be reported to Westinghouse in any form. All errors shall be evaluated and documented consistent with the Westinghouse Quality Management System and the information required by EXHIBIT 11-1 EXCEPTION REPORT. The Platform Lead shall report errors to all users by issuing Technical Bulletins in accordance with Westinghouse Level II Policies and Procedures (Reference 4). If a receipt is needed from the customer or verification that some site activities have occurred, then a formal reply shall be requested in the Technical Bulletin. When the error impacts protection and/or important-to-safety class software or protection system designs using the software, then the user is responsible for documenting appropriate action as necessary, including 10CFR21 evaluations.

11.4 CORRECTIVE ACTION

The EPM shall establish as a clear objective the goal of resolving all validation test problems (via Exception Reports), verification review comments, and other reported errors expeditiously to minimize the potential for unidentified effects during later life cycle phases.

The corrective action procedures used shall be based on the level of problem reported. Problems that may require a process improvement to prevent reoccurrence or problems that affect interfaces between workgroups may require management attention and follow up activities. These types of problems shall be entered into the Westinghouse Corrective Actions Process in accordance with Reference 4.

In addition, the EPM shall adhere to the following corrective action methodology that:

- Problems are identified, evaluated, documented and, if required, corrected by the appropriate reporting mechanism (Sections 11.1 and 11.2).

- Corrections or changes shall be controlled in accordance with the SCMP (subsection 6.3.2).

- Preventive actions and corrective actions are documented on the appropriate form and distributed to the appropriate NA groups.

(Last Page of Section 11)

SECTION 12
SECURE DEVELOPMENT AND OPERATIONAL ENVIRONMENT PLAN

12.1 INTRODUCTION

12.1.1 Overview

This plan addresses computer security throughout the life cycle phases of a Common Q™ safety system and summarizes the quality standards and design control measures that provide a secure development and operational environment, and provides the means for the system to be designed for high functional reliability commensurate for safety. The development phases include the concept, requirements, design, implementation and testing, as defined in subsection 1.4.1.

12.2 LIFE CYCLE PHASE ACTIVITIES

12.2.1 Concept Phase

12.2.1.1 Secure Operational Environment Capabilities

[

]a,c,e

[

]a,c,e

12.2.1.2 Secure Development Environment

[

]a,c,e

12.2.1.2.1 General Life cycle Vulnerabilities

[

]a,c,e

[

]a,c,e

[

]a,c,e

12.2.1.2.2 Isolated Development Infrastructure

[

]a,c,e

[

]a,c,e

12.2.1.3 Outputs from the Concept Phase

If contracted by the licensee, the output from the Concept Phase is a concept phase secure operational environment assessment that provides input into the requirements phase.

12.2.2 Requirements Phase

12.2.2.1 System Features – Security Functional Performance Requirements

[

]a,c,e

12.2.2.2 System Requirements Independent Verification & Validation (IV&V)

[

]a,c,e

12.2.2.3 Requirements Phase Outputs

The outputs of this phase are the incorporation of the secure operational environment requirements into the system requirements documents and completion of the requirements phase IV&V.

12.2.3 Design Phase

[

]a,c,e

12.2.3.1 Design Phase Outputs

The outputs of this phase are:

- Software design documentation.
- IV&V Phase Summary Report.

12.2.4 Implementation Phase

In the software implementation phase, the executable code modules are created. The application modules are integrated with platform software to produce code that is downloaded into Common Q™ processors for IV&V testing.

[

]a,c,e

[

]a,c,e

12.2.4.1 Implementation Phase Outputs

The outputs of this phase are:

- Software Release Records.
- IV&V Phase Summary Report.

12.2.5 Testing Phase

[

]a,c,e

12.2.5.1 Testing Phase Outputs

The outputs of this phase are:

- Test Reports.
- IV&V Phase Summary Report.

(Last Page of Section 12)

SECTION 13
EXHIBITS

This section contains the following Exhibits:

EXHIBIT 1-1 RELATIONSHIP OF SPM TO IEEE STANDARDS

EXHIBIT 2-1 DESIGN/IV&V TEAM ORGANIZATION

EXHIBIT 4-1 ASSIGNMENT OF COMMON Q™ SOFTWARE TO CLASSES

EXHIBIT 4-2 COMMON Q™ SOFTWARE DEVELOPMENT PROCESS

EXHIBIT 4-3 TASKS REQUIRED FOR SOFTWARE CATEGORIES

EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES

EXHIBIT 5-2 CHECKLIST NO. 1, SOFTWARE VERIFICATION AND VALIDATION CONCEPT PHASE CHECKLIST

EXHIBIT 5-3 CHECKLIST NO. 2, SOFTWARE VERIFICATION AND VALIDATION REQUIREMENTS PHASE CHECKLIST

EXHIBIT 5-3 CHECKLIST NO. 2, SOFTWARE VERIFICATION AND VALIDATION REQUIREMENTS PHASE CHECKLIST

EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION DESIGN PHASE CHECKLIST

EXHIBIT 5-5 CHECKLIST NO. 4, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST

EXHIBIT 5-6 CHECKLIST NO. 5, SOFTWARE VERIFICATION AND VALIDATION TEST PHASE CHECKLIST

EXHIBIT 5-7 CHECKLIST NO. 6, SOFTWARE VERIFICATION AND VALIDATION INSTALLATION AND CHECKOUT PHASE CHECKLIST

EXHIBIT 5-8 IEEE STANDARD 1012-2004 COMPLIANCE TABLE

EXHIBITS (cont.)

EXHIBIT 1-1 RELATIONSHIP OF SPM TO IEEE STANDARDS

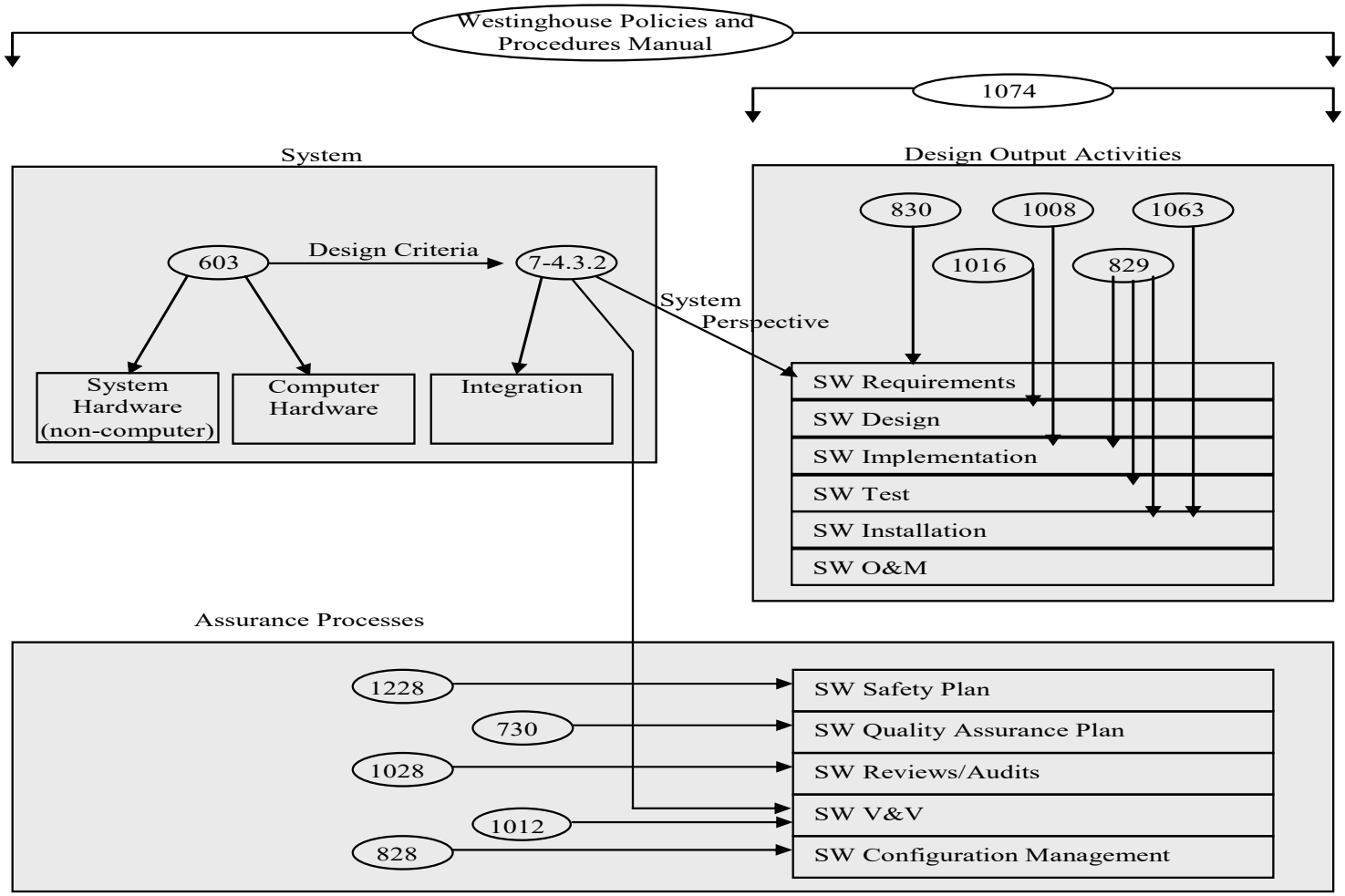
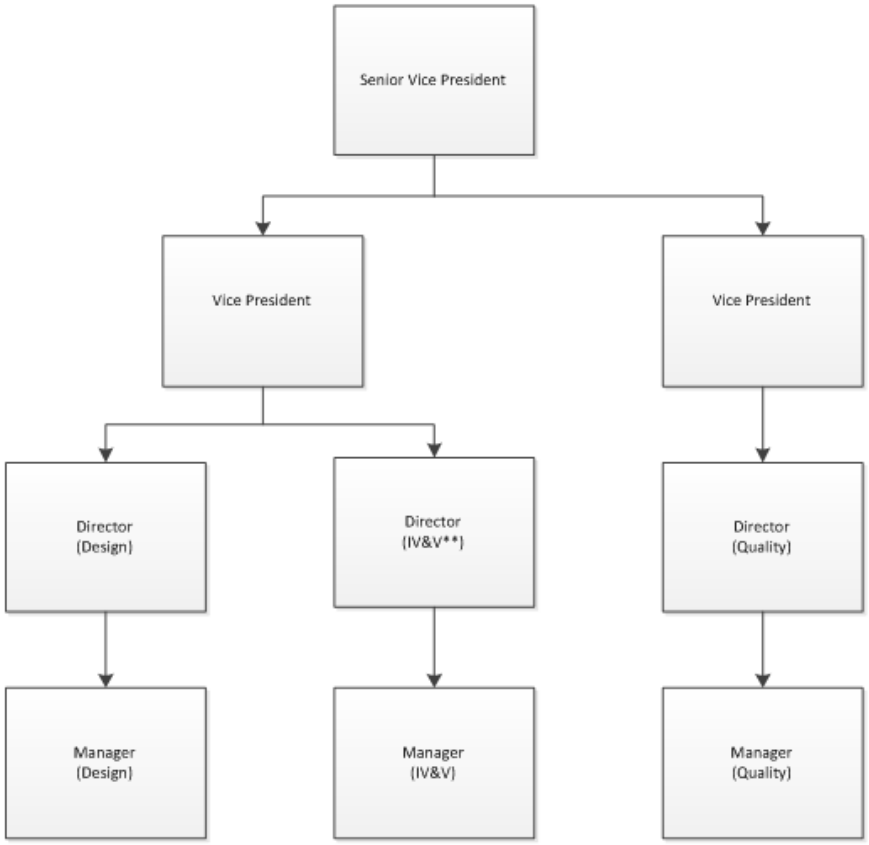


EXHIBIT 2-1 DESIGN/IV&V TEAM ORGANIZATION*



*This example organization chart shows the minimum level of separation required for the Design, IV&V, and Quality Teams

**System level validation testing can be performed by another group, which meets the same level of independence as the IV&V group depicted in this organization chart

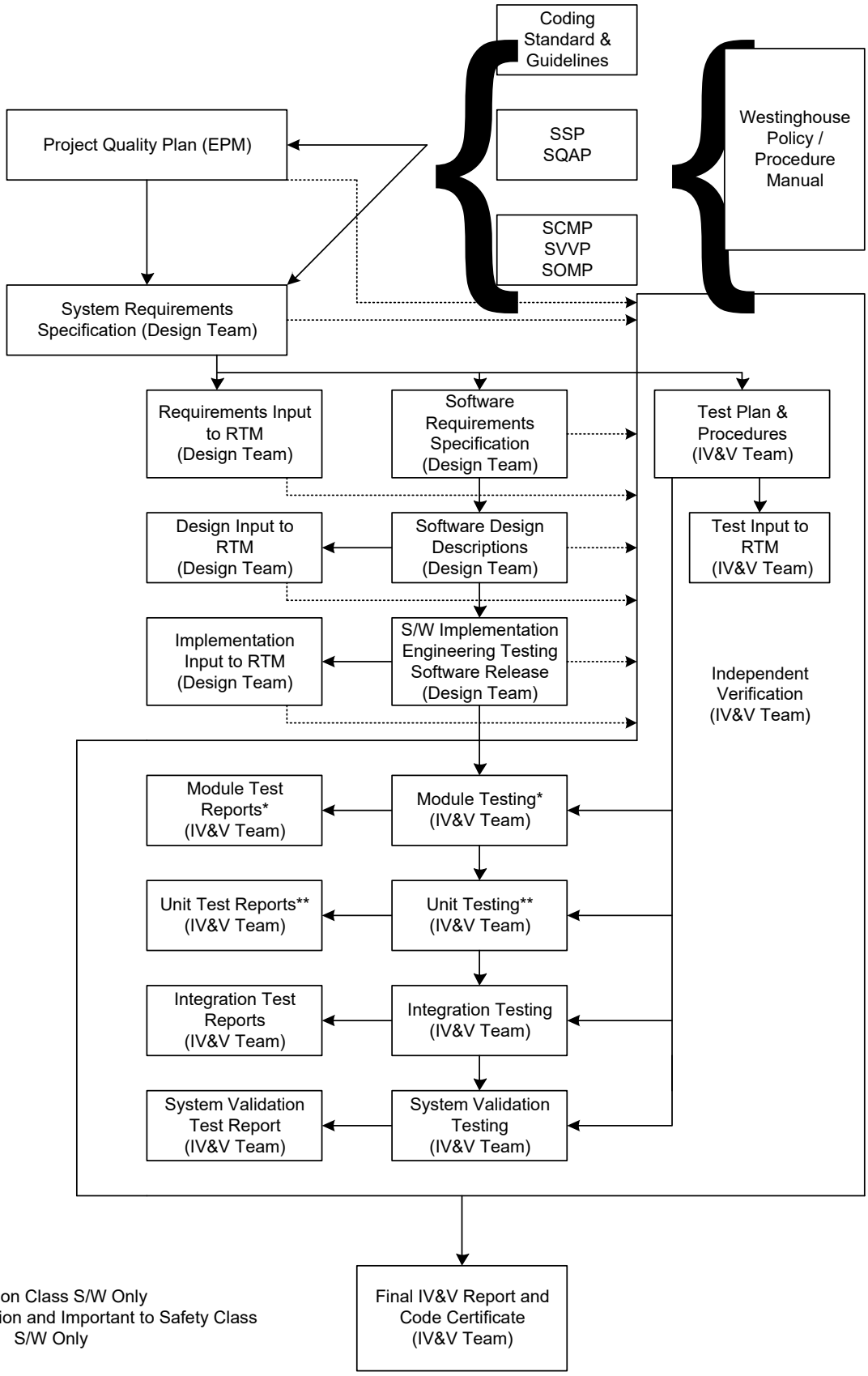
EXHIBIT 4-1 ASSIGNMENT OF COMMON Q™ SOFTWARE TO CLASSES

SYSTEM	SUB-SYSTEM SCOPE	CLASS
Plant Protection/Reactor Protection	Safety Critical Kernel (LCL, Bistable)	Protection
	Maintenance and Test Panel (MTP)	Important-to-Safety
	Operator's Module	Important-to-Safety
	Interface and Test Processor (ITP)	Important-to-Safety
	Intra-Divisional Communication Software (AF100)	Important-to-Safety
	All Other Software	General Purpose
	Development Tools	General Purpose
Engineered Safety Features Actuation	Safety Critical Kernel (ILP)	Protection
	MTP	Important-to-Safety
	Intra-Divisional Communication Software (AF100)	Important-to-Safety
	All Other Software	General Purpose
	Development Tools	General Purpose

EXHIBIT 4-1 ASSIGNMENT OF COMMON Q™ SOFTWARE TO CLASSES (cont.)

SYSTEM	SUB-SYSTEM SCOPE	CLASS
Core Protection Calculator	Safety Critical Kernel (FLOW, UPDATE, POWER, STATIC)	Protection
	CEAC Software	Protection
	MTP	Important-to-Safety
	Operators Module	Important-to-Safety
	Intra-Divisional Communication Software (AF100)	Important-to-Safety
	CEAPDS	Important-to-Availability
	All Other Software	General Purpose
	Development Tools	General Purpose
Post Accident Monitoring	Kernel Software (CET, SM, RVL monitoring)	Important-to-Safety
	Flat Panel Display System	Important-to-Safety
	Intra-Divisional Communication Software (AF100)	Important-to-Safety
	All Other Software	General Purpose
	Development Tools	General Purpose

EXHIBIT 4-2 COMMON Q™ SOFTWARE DEVELOPMENT PROCESS



* Protection Class S/W Only
** Protection and Important to Safety Class S/W Only

EXHIBIT 4-3 TASKS REQUIRED FOR SOFTWARE CATEGORIES

TASK	ORIGINAL SOFTWARE	ETBM SOFTWARE	ENM SOFTWARE
SQA PLANNING PHASE			
SOFTWARE QUALITY ASSURANCE PLAN	X	X	X
CODING STANDARDS	X	X	
SOFTWARE VERIFICATION AND VALIDATION PLAN	X	X	X
SOFTWARE CONFIGURATION MANAGEMENT PLAN	X	X	X
SOFTWARE REQUIREMENTS PHASE			
SYSTEM REQUIREMENTS	X	X	X
PROTOTYPE CODING	As Required		
SOFTWARE REQUIREMENTS	X	X	X
SOFTWARE DESIGN PHASE			
SOFTWARE DESIGN DESCRIPTION	X	X	
REQUIREMENTS TRACEABILITY ANALYSIS	X	X	X
SOFTWARE IMPLEMENTATION PHASE			
MODULE CODING	X	X	
TEST PLAN	X	X	X
MODULE TEST PROCEDURE (Protection)	X	X	
MODULE TEST EXECUTION	X	X	

EXHIBIT 4-3 TASKS REQUIRED FOR SOFTWARE CATEGORIES (cont.)

TASK	ORIGINAL SOFTWARE	ETBM SOFTWARE	ENM SOFTWARE
MODULE TEST EXECUTION REPORT (Protection)	X	X	
UNIT TEST PROCEDURE (Protection and Important-to-Safety)	X	X	X
UNIT TEST EXECUTION	X	X	X
UNIT TEST REPORT (Protection and Important-to-Safety)	X	X	X
REQUIREMENTS TRACEABILITY ANALYSIS	X	X	
SOFTWARE TEST PHASE			
INTEGRATION TEST PROCEDURE	X	X	X
INTEGRATION TEST EXECUTION	X	X	X
INTEGRATION TEST REPORT	X	X	X
SYSTEM VALIDATION TEST PROCEDURE	X	X	X
SYSTEM VALIDATION TEST EXECUTION	X	X	X
SYSTEM VALIDATION TEST REPORT	X	X	X
FACTORY ACCEPTANCE TEST PROCEDURE	X	X	X
FACTORY ACCEPTANCE TEST EXECUTION	X	X	X
FACTORY ACCEPTANCE TEST REPORT	X	X	X
USER DOCUMENTATION	X	X	X
SOFTWARE IV&V REPORT	X	X	X

EXHIBIT 4-3 TASKS REQUIRED FOR SOFTWARE CATEGORIES (cont.)

TASK	ORIGINAL SOFTWARE	ETBM SOFTWARE	ENM SOFTWARE
SOFTWARE INSTALLATION & CHECKOUT PHASE			
INSTALLATION TEST (SAT) PROCEDURE*	X	X	X
INSTALLATION TEST (SAT) EXECUTION*	X	X	X
INSTALLATION TEST (SAT) REPORT*	X	X	X
SOFTWARE OPERATION AND MAINTENANCE PHASE			
MAINTAIN SOFTWARE	X	X	X
SOFTWARE RETIREMENT PHASE			
RETIREMENT NOTIFICATION	X	X	X

ETBM – Existing Software To Be Modified
ENM – Existing Software Not To Be Modified

*Applicable if within Westinghouse scope of supply.

EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES

TASK	PROTECTION	IMPORTANT-TO-SAFETY	IMPORTANT-TO-AVAILABILITY	GENERAL PURPOSE
SOFTWARE REQUIREMENTS PHASE				
SYSTEM AND SOFTWARE REQUIREMENTS	DT/VT	DT/VT	DT	DT
REQUIREMENTS VERIFICATION	VT	VT***	N/A	N/A
SOFTWARE DESIGN PHASE				
SOFTWARE DESIGN DESCRIPTION	DT/VT	DT/VT	DT	DT
PROTOTYPE CODING	DT	DT	DT	DT
DESIGN VERIFICATION	VT	VT***	N/A	N/A
SOFTWARE IMPLEMENTATION PHASE				
TEST PLAN (MAY BE PART OF SVVP)	VT	VT	DT	DT
MODULE CODING	DT/VT	DT/VT	DT	DT
MODULE TEST PROCEDURE**	VT	N/A	N/A	N/A
MODULE TEST EXECUTION/REPORT**	VT	N/A	N/A	N/A
UNIT TEST PROCEDURE	VT	VT	N/A	N/A
UNIT TEST EXECUTION/REPORT	VT	VT	N/A	N/A
IMPLEMENTATION VERIFICATION	VT	VT***	DT	DT
SOFTWARE TEST PHASE				
INTEGRATION TEST PROCEDURE	VT	VT	DT	DT

EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES (cont.)

TASK	PROTECTION	IMPORTANT-TO-SAFETY	IMPORTANT-TO-AVAILABILITY	GENERAL PURPOSE
INTEGRATION TEST EXECUTION	VT	VT	DT	DT
INTEGRATION TEST REPORT	VT	VT	DT	DT
SYSTEM VALIDATION TEST PROCEDURE	VT	VT	DT	DT
SYSTEM VALIDATION TEST EXECUTION	VT	VT	DT	DT
SYSTEM VALIDATION TEST REPORT	VT	VT	DT	DT
FACTORY ACCEPTANCE TEST PROCEDURE	VT	VT	DT	DT
FACTORY ACCEPTANCE TEST EXECUTION	VT	VT	DT	DT
FACTORY ACCEPTANCE TEST REPORT	VT	VT	DT	DT
USER DOCUMENTATION	DT/VT	DT/VT	DT	DT
SOFTWARE IV&V REPORT	VT	VT	N/A	N/A
SOFTWARE INSTALLATION & CHECKOUT PHASE				
INSTALLATION TEST (SAT) PROCEDURE*	VT	VT	DT	DT
INSTALLATION TEST (SAT) EXECUTION*	VT	VT	DT	DT
INSTALLATION TEST (SAT) REPORT*	VT	VT	DT	DT
KEY: ACTIVITY PERFORMED BY/REQUIRES IV&V (i.e., DT/VT means DT performs activity and requires IV&V, DT means DT performs activity but does not require IV&V, VT means activity performed by IV&V, and N/A means	DT = DESIGN TEAM		VT = IV&V TEAM	

TASK	PROTECTION	IMPORTANT-TO-SAFETY	IMPORTANT-TO-AVAILABILITY	GENERAL PURPOSE
activity is not required)				

*Applicable if in Westinghouse scope of supply.

** These activities are performed for Protection Class software only.

*** Same IV&V activities as Protection Class software except for Software Hazards Analysis described in subsection 3.4.1.

EXHIBIT 5-2 CHECKLIST NO. 1, SOFTWARE VERIFICATION AND VALIDATION CONCEPT PHASE CHECKLIST

Software Item Name: _____ Software Item ID: _____

		YES
1.	Were the following IV&V tasks completed during the Concept Phase?:	
a.	Review Concept documents for consistency, incompatibilities, and compliance to regulations.	___
b.	Identify major constraints of interfacing systems.	___
c.	Identify constraints or limitations of proposed system.	___
d.	Assess criticality of each software item.	___
e.	Configuration management evaluation of all applicable conceptual documents (including evaluating if conceptual documents have been captured properly and placed under configuration control).	___
f.	Verify tracing of project baseline documents for compliance to customer requirements, applicable product documents and regulatory standards and guidelines.	___
g.	Complete EXHIBIT 5-2 CHECKLIST NO. 1, SOFTWARE VERIFICATION AND VALIDATION CONCEPT PHASE CHECKLIST and reference completed checklist in the Concept Phase IV&V Report.	____

Reviewer's comments (Optional): _____

_____

_____

_____

Reviewed by: Name _____ Signature _____ Date _____

EXHIBIT 5-3 CHECKLIST NO. 2, SOFTWARE VERIFICATION AND VALIDATION REQUIREMENTS PHASE CHECKLIST

Software Item Name: _____ Software Item ID: _____

		YES
1.	Were the following IV&V tasks completed during the Requirements Phase?:	
	a. Review the adequacy and accuracy of the RTM as prepared by the design team. The review shall include verification that all functional, hardware interface, software, performance, and user requirements have been included.	___
	b. Assess allocation of functions to hardware and software items	___
	c. Perform or review the adequacy and accuracy of the following software safety analyses using Reference 26, Annex A.1 as criteria: 1. Criticality 2. Specification 3. Timing and Sizing 4. Different software systems (if applicable)	___
	d. Review applicable Commercial Grade Dedication reports to evaluate the suitability of the commercially dedicated item for the particular implementation being verified. Commercial Grade Dedication Report characteristics are defined in subsection 5.5.3.2, item 8.	___
	e. Verify identification of the original software items developed under this SPM for generic application that will be used in the project; verify that the qualification status has been identified and is appropriate; and verify through the RTA process that this software meets the requirements.	___
	f. Develop a Common Q™ specific test plan in accordance with subsection 4.3.2.2. Verify that it includes the following topics as a minimum: 1. General approach including: identification of test procedures, general test methods, documentation of results, and traceability methods to the SRS and SDD.	

2. Requirements for testing including: test boundary conditions on inputs and unexpected input conditions.
3. Test management including: personnel, resources, organization, and responsibilities.
4. Procedures for qualification and control of the hardware to be used in testing.
5. Qualification and use of software tools.
6. Installation test requirements for existing software that is used without modification.
7. Regression test requirements for previously qualified software to be modified.
8. Delineate major features of the system that will be tested.

___

g. Configuration Management Evaluation – assess the applicability of the Software Configuration Management Plan (Section 6) to the project as augmented by the project plan. ___

h. A review shall be conducted to verify that each hazard identified in the software hazard analysis and/or failure modes and effects analysis, has been mitigated or the risks associated with the hazard have been reduced to an acceptable level. ___

i. Complete EXHIBIT 5-3 CHECKLIST NO. 2, SOFTWARE VERIFICATION AND VALIDATION REQUIREMENTS PHASE CHECKLIST and reference completed checklist in the Requirements Phase IV&V Report. ___

2. Does the review of available documentation identify: YES

a. Completeness and correctness in specifying the performance requirements and operational capabilities and concepts of the system. Does the system design implement the functional requirements, are the plant parameters defined in the functional design being monitored in the system design? ___

b. Completeness and correctness in system definition and interfaces with other equipment. Perform analysis of requirements decomposition – are subsystems defined with interface requirements noted? ___

c. Unambiguous, correct, and consistent description of the interfaces and performance characteristics of each major function. ___

d. Establishment of a reasonable and achievable set of test requirements. These requirements should be related to performance goals and also define acceptability criteria. ___

e. Definition of physical characteristics, reliability and maintainability objectives, operating environment, transportability constraints, design and documentation standards. ___

f. Definition of the necessary training requirements and considerations. ___

g. Treatment of man/machine interface requirements. ____

h. Definition of integration requirements. ____

i. Definition of installation, operation, and maintenance requirements. ____

j. Are the bases for the requirements identified? ____

k. Review requirements with respect to the following possible errors: ____

1. Inadequate or partially missing performance criteria.
2. Inadequate or partially missing operating rules (or information).
3. Inadequate or partially missing ambient environment information.
4. Requirements that are incompatible with other requirements.
5. Inadequate or partially missing system mission information.
6. Ambiguous or requirements subject to misinterpretation.
7. User's needs not properly understood or reflected.
8. Requirement not traceable to user's needs.
9. Requirements which cannot be physically tested.
10. Accuracy specified does not conform to the need.
11. Data environment inadequately described.
12. Input/output data parameters units incorrect.
13. Erroneous external interface definition.
14. Initialization of the system not properly considered.
15. Vague requirements of the functions to be performed.
16. Required processing inaccurate.
17. Required processing inefficient.
18. Required processing not necessary.
19. Missing requirements on flexibility, maintainability.
20. Missing or incomplete requirements of response to abnormal data or events.

21. Inadequate or incorrect algorithm.
22. Incorrect timing/synchronization requirements.
23. Incorrect hardware interface requirements.
24. Incorrect allocation of system resources.

l. Has the document author updated the RTM? Is the RTM adequate and accurate in providing the traceability of requirements? ____

3. Do the software/hardware interface requirements identify: YES

a. All input/output and requirements, including range, accuracies and data rates. ___

b. Design features (e.g., keylocks) which provide administrative control of all devices capable of changing the content of the stored programs or data. ____

c. Initialization requirements, such as power-up and power-down. ___

d. Design features for the detection of system failures (e.g., on-line self-tests). ___

e. Manually-initiated in-service test or diagnostic capabilities. ___

f. Human factors engineering design features that ease the interaction with the system for operation, maintenance, and testing. ___

g. Margins for timing, memory/buffer size, etc., including minimum margins for design. ___

h. Interrupt features. ___

4. Do the software requirements identify: YES

a. Process inputs including voltage and sampling frequency. ___

b. System software, utility routines and other auxiliary programs required for operation ___

c. Algorithms to be programmed with consideration to handling of abnormal events ___

d. Data files and data required for the algorithms, including symbolic names and requirements for flexibility. ___

e. Process outputs, including ranges, accuracies, update interval, and human factors considerations of the operator interface. ___

f. Initialization requirements, such as initial values and start-up sequence. ___

g. Parameters to configure system program logic for response to detected failures. ___

h. Operator interface requirements (switches, readouts). ___

i. In-service test or diagnostic capabilities. ___

j. Timing requirements for all time-dependent events, including overall system requirements. ___

k. Limitations on processor time and memory capabilities. ___

l. Security requirements (e.g., passwords). ___

Reviewer's comments (Optional): _____

_____

_____

_____

Reviewed by: Name _____ Signature _____ Date _____

EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION DESIGN PHASE CHECKLIST

Software Item Name: _____ Software Item ID: _____

		YES
1.	Were the following IV&V tasks completed during the Design Phase?:	
a.	Review system design documentation to verify the system design completely and correctly performs the functions specified in the requirements documents.	___
b.	Review system design documentation to determine that the hardware/software interface design specifications are understandable, unambiguous, reasonable, implementable, accurate, complete, and are a faithful translation of the hardware/software interface design requirements into hardware/software interface design specifications	___
c.	Review software design documentation to verify design requirements are adequately incorporated. The design documentation shall address all software requirements and provide a correlation of the design elements with the software requirements.	___
d.	Perform or review the adequacy and accuracy of the following software safety design analyses using Reference 26, Annex A.2 as criteria: a. Logic b. Data c. Interface d. Constraint e. Functional f. Software element	___
e.	Review current criticality analysis assessment for continued applicability.	___
f.	Perform the Requirements Traceability Analysis.	___
g.	Configuration Management – Confirm that the verified design documents have	

been properly placed under configuration control. ___

h. Begin preparing module, unit, integration, system validation and FAT test procedures in accordance with Section 5.8. ___

i. Review the software hazard analysis and/or failure modes and effects analysis to verify that any new hazards have been documented during this phase. ___

j. Complete EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION DESIGN PHASE CHECKLIST and reference in the Design Phase IV&V Report. ___

2. Does the available documentation adequately address: YES

a. Architecture, for both hardware and software. ___

b. Input/output interface. ___

c. System and Executive Control. ___

d. Operating Sequences – initialization, start-up, error detection, restart, etc. ___

e. Testability – use of test equipment, such as data tapes, simulations, etc. ___

f. Timing analysis – sampling rates, response time, etc. ___

g. Availability – what does analysis and data indicate? ___

h. Algorithm design and data verification. ___

i. Information flow – communication between subsystems, data management and signal conversion to engineering units. ___

j. Human factors engineering. ___

k. Is the design correct, complete, and traceable to requirements? Has the document author updated the RTM? Is the RTM adequate and accurate in providing the traceability of software design descriptions to requirements? ___

l. Is the design internally consistent? ___

m. Is the design feasible? ___

n. Is the design clear and unambiguous? ___

o. Is the design testable? ___

p. Software design as a whole emphasizing allocation of software components to function, functional flows, storage requirements and allocations, and design of the database. ___

q. General description of the size and operating characteristics of all support software. ___

Reviewer's comments (Optional): _____

_____

_____

_____

Reviewed by: Name _____ Signature _____ Date _____

EXHIBIT 5-5 CHECKLIST NO. 4, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST

Software Item Name: _____ Software Item ID: _____

		YES
1.	Were the following IV&V tasks completed during the Implementation Phase?:	
a.	Review the as-built software documentation to verify the as-built software completely and correctly implements the design specified in the system design documents.	___
b.	Perform or review the adequacy and accuracy of following software safety code analyses using Reference 26, Annex A.3 as criteria: 1. Logic 2. Data 3. Interface 4. Constraint 5. Programming Style 6. Non-critical code 7. Timing and sizing	___
c.	Review current criticality analysis assessment for continued applicability.	___
d.	Perform the Requirements Traceability Analysis.	___
e.	Evaluate Software Configuration Management activities and verify the requirements of Section 6 are fulfilled.	___
f.	Hardware implementation review is normally conducted as part of the hardware quality assurance activities defined elsewhere	___
g.	For Protection Class software, review software testing records to verify adequate structural testing.	___
h.	Integration, System validation and FAT test procedures are prepared in accordance with Section 5.8, based upon the requirements of the design and shall include test	

cases encompassing the range of usage intended for the system. Test Procedure shall include the characteristics listed in subsection 5.5.5.2, item 11. ___

i. Review the software hazard analysis and/or failure modes and effects analysis to verify that any new hazards have been documented during this phase. ___

j. Complete EXHIBIT 5-5 CHECKLIST NO. 4, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST and reference completed checklist in the Implementation Phase IV&V Report. ___

2. Review the source code with respect to the following: YES

a. Does the source code conform to specified standards and procedures including internal proprietary information handling and coding standards and guidelines? ___

b. Are the comment statements sufficient to give an adequate description of each routine? ___

c. Is the source code clearly understandable? ___

d. Is the source code logically consistent with design specs? Has the programmer updated the RTM? Is the RTM adequate and accurate in providing the traceability of software modules to software design descriptions? ___

e. Are all variables properly specified and used? ___

f. Is there satisfactory error checking? ___

g. Do all subroutine calls transfer variables correctly? ___

h. Is the data read in each file consistent with the data written to it? ___

3. Do the database modules adequately and correctly reflect: YES

a. Program and general content. ___

b. File organization, layout, and residence. ___

c. File accessing methods. ___

d. File size. ___

e. Data record description(s) – record layout, field allocations, field names, detailed description of field contents. ___

f. Initialization requirements. ___

g. Data accuracy. (See subsection 5.4.5.4.) ___

h. Data completeness. (See subsection 5.4.5.4.) ___

i. Maintenance. ___

4. Review Module Test Documentation YES

a. Has module testing been documented for all protection class software? ___

b. Is the test coverage documented? ___

c. Is the test coverage adequate? Verify that all branches of all software modules have been tested or that adequate justification and analysis has been completed for untested branches. ___

d. Do module test reports indicate correct execution of critical software elements? ___

5. Review Unit Test Documentation YES

a. Has unit testing been documented for all protection and important-to-safety class software? ___

b. Is the test coverage documented? ___

c. Is the test coverage adequate? Verify that all functions of all software units have been tested or that adequate justification and analysis has been completed for untested functions. ___

d. Do unit test reports indicate correct execution of critical software functions? ___

6. Do procedures exist (as necessary) to: YES

a. Generate all object code required for system generation and produce the corresponding software listings. ___

b. Generate a customized database and system parameter file according to plant-specific requirements and produce the corresponding listings. ___

c. Configure the operating system according to the plant-specific hardware configuration. ___

d. Generate the system from the above results. ___

e. Initialize and boot the system after system generation. ___

f. Modify, enhance, and maintain the system including the usage of diagnostic and debugging utilities. ___

g. Generate and update displays. ___

h. Integrate the hardware/configured software. ___

Reviewer's comments (Optional): _____

_____

_____

_____

Reviewed by: Name _____ Signature _____ Date _____

EXHIBIT 5-6 CHECKLIST NO. 5, SOFTWARE VERIFICATION AND VALIDATION TEST PHASE CHECKLIST

Software Item Name: _____ Software Item ID: _____

		YES
1.	Were the following IV&V tasks completed during the Test Phase?:	
	a. Follow up on changes and corrections made in the system in accordance with change control procedures in Section 6.	___
	b. Perform the Requirements Traceability Analysis.	___
	c. Review user documentation. This may be done as part of the Installation and Checkout phase if within Westinghouse's scope of supply by specific contract.	___
	d. Perform Functional Review to verify that all requirements specified in the SRS have been met. This review shall include an overview of all documentation and a review of the results of the previous reviews, including Software Requirements Review, ADR, CDR, and if applicable, interim IV&V reports (for Protection and Important-to-Safety class software). The tasks conducted in this phase meet the requirements of subsection 4.6.2.5, Functional Review.	___
	e. Complete EXHIBIT 5-6 CHECKLIST NO. 5, SOFTWARE VERIFICATION AND VALIDATION TEST PHASE CHECKLIST and reference completed checklist in the Installation and Checkout Phase IV&V Report.	___
	f. At the completion of all other tasks in this phase, a final IV&V report is issued. The final IV&V report may not be issued until the Installation and Checkout Phase if within Westinghouse's scope of supply by specific contract. Final IV&V report characteristics are defined in subsection 5.5.7.2, item 4.	___
2.	Verify program integration with hardware in accordance with the following:	YES
	a. Does the integrated program conform to the maximum resource requirements for memory size and program execution time?	___

b. Does the integrated program interface properly with external files? ___

c. Have all of the elements of the integrated program been identified in the module list? ___

d. Does the code compile and link without errors? ___

e. Are interfaces between programs, data files, and libraries correctly programmed? ___

3. Verify program validation in accordance with the following: YES

a. Has the test engineer updated the RTM? Is the RTM adequate and accurate in providing the traceability of software test cases to software modules and requirements? ___

b. Has each section of the test procedure been completed accurately? ___

c. Have all tests passed and have all requirements of testing been fulfilled? ___

d. Have applicable software hazard prevention and/or control features been tested? ___

4. Verify test results and report in accordance with the following: YES

a. Does the Test Report comply with the format specified in the Test Plan? ___

– Does it provide complete identification of the program tested? ___

– Does it specify the scope of the Test Report? ___

– Does it reference the Test Plan and any other relevant documents? ___

– Does it include a complete and accurate description of the test environment:

Hardware configuration? ___

Support software used? ___

– Does it describe and justify each deviation from the Test Plan? ___

– Does it provide a summary of test results? ___

– Does it include an evaluation of the program performance with respect to requirements? ___

– Does it provide recommendations for retesting, or program acceptance, or both? ___

– Does it provide a detailed description of the results of each test case? ___

– Does it include a copy of the test case log? ___

– Does it include all discrepancy reports prepared during the testing? ___

b. Is the information in the Test Report an accurate statement of the testing performed? ___

– Does the output summary of test results accurately reflect the test output produced? ___

– Is the evaluation of the program a realistic and accurate reflection of the test results? ___

– Are the recommendations regarding retesting and acceptance sound and based on the test results? ___

– Do the descriptions of the test case results accurately reflect actual test outputs? ___

– Is the test case log complete and consistent with actual test output? ___

– Are the discrepancy reports complete and consistent with actual test output? ___

c. Have all test cases been executed correctly?

– Does the test case log indicate performance of each test case in the specified test environment using specified test procedures? ___

– Is there an explanation for any deviation from the specified test environment or procedures? ___

– Is there an Exception Report for each deviation from expected results? ___

– Were correct input data used for each test case? ___

– Is the output produced by each test case accurately reported? ___

5. General Assessment Questions: YES

a. Is there convincing evidence that the system meets protection system safety requirements? ___

b. Is there convincing evidence that the system does not introduce any new hazards? ___

Reviewer's comments (Optional): _____

_____

_____

_____

Reviewed by: Name _____ Signature _____ Date _____

EXHIBIT 5-7 CHECKLIST NO. 6, SOFTWARE VERIFICATION AND VALIDATION INSTALLATION AND CHECKOUT PHASE CHECKLIST

Software Item Name: _____ Software Item ID: _____

		YES
1.	Were the following IV&V tasks completed during the Installation and Checkout Phase?:	
	a. Review installation procedures and user manuals to verify that they are complete and correct.	___
	b. Review training materials (if within Westinghouse's scope of supply) for the following: 1. Safety training for the users, operators, maintenance and management personnel 2. System startup training 3. Safety training requirements are met	___
	c. Review that the Exception Report Log that was maintained in accordance with the SAT plan.	___
	d. Configuration Management - Evaluate that the manuals and procedures have been properly placed under configuration control.	___
	e. Complete EXHIBIT 5-7 CHECKLIST NO. 6, SOFTWARE VERIFICATION AND VALIDATION INSTALLATION AND CHECKOUT PHASE CHECKLIST and reference completed checklist in Final IV&V report.	___
	f. At the completion of all others tasks in this phase, prepare and issue the final IV&V report. This report will be issued during the Test Phase if the Installation and Checkout Phase are not within Westinghouse's scope of supply.	___
2.	Is the user documentation installation package sufficient to install the software on the delivered hardware?	___
3.	Is the user documentation clear, unambiguous, and consistent with system requirements?	___
4.	Does the IV&V report have positive findings?	___

5. Have all discrepancies and IV&V findings been resolved to the satisfaction of the IV&V team? ___

6. Are SCM controls in place for the user to report errors? ___

7. Training documentation meet Safety Training Requirements ___

8. Is the software installed correctly? ___

9. Have configuration tables been correctly initialized, if such are used? ___

10. Are operating documents present, correct, complete, and consistent? ___

Reviewer's comments (Optional): _____

_____

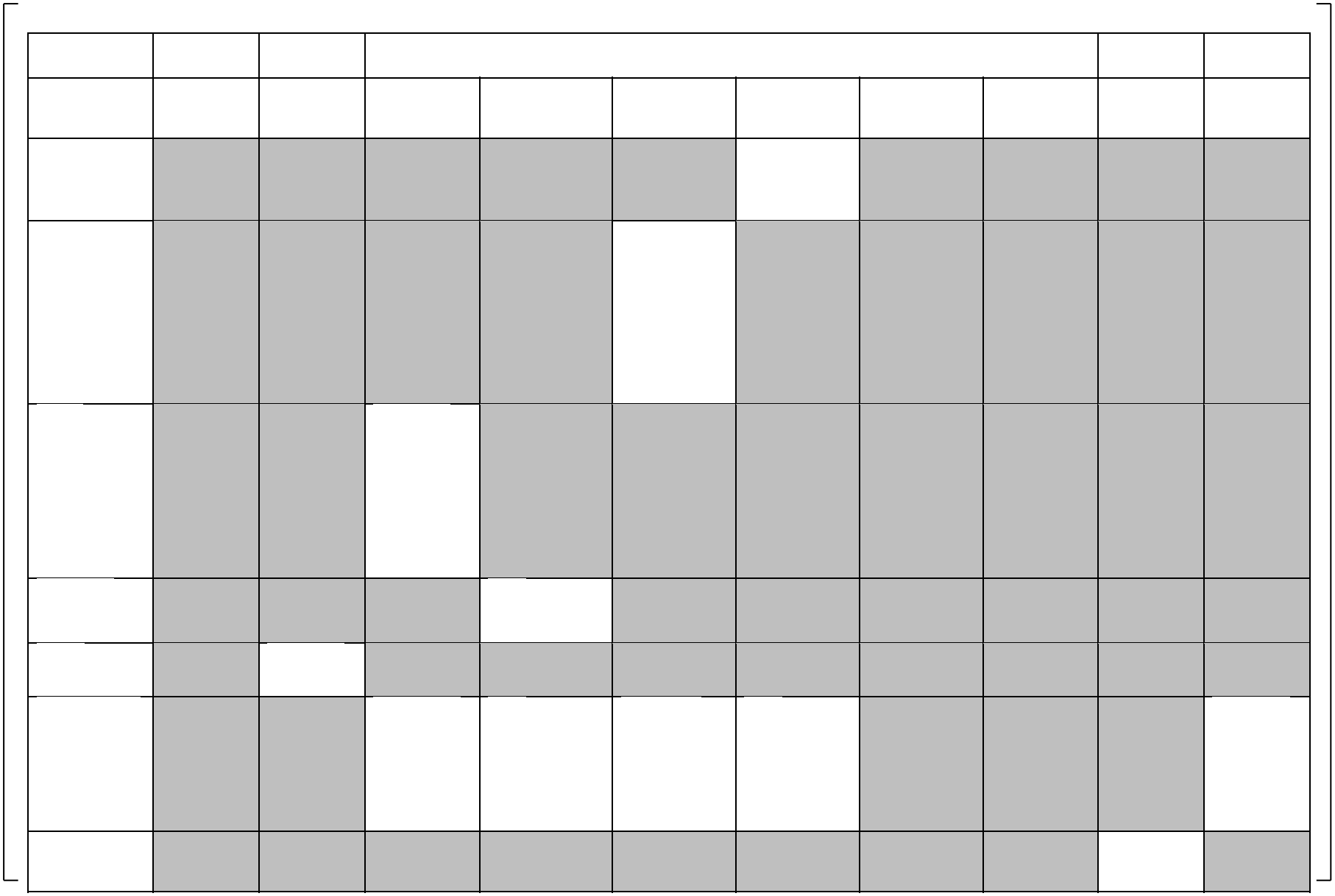
_____

Reviewed by: Name _____ Signature _____ Date _____

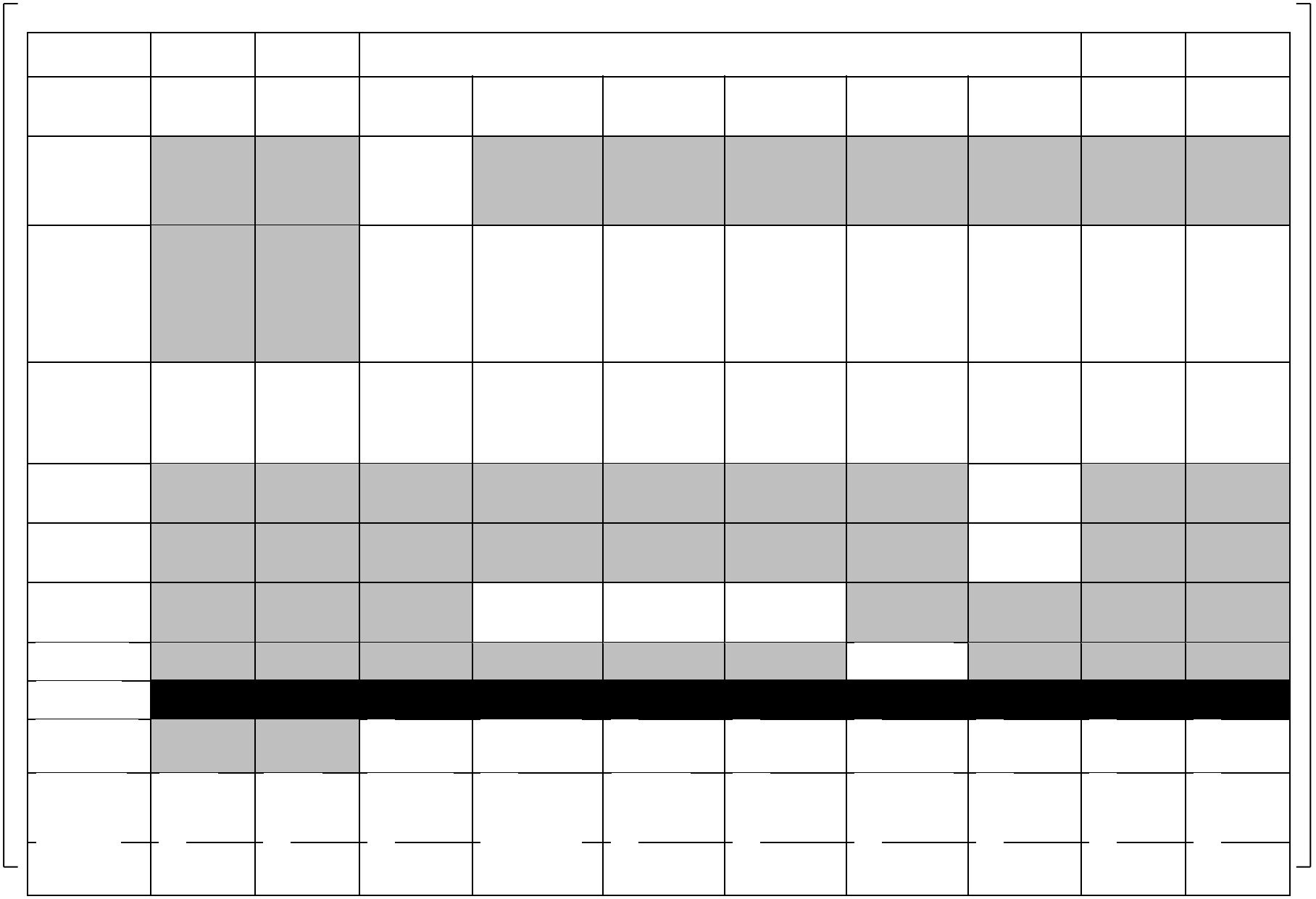
EXHIBIT 5-8 IEEE STANDARD 1012-2004 COMPLIANCE TABLE

a,c

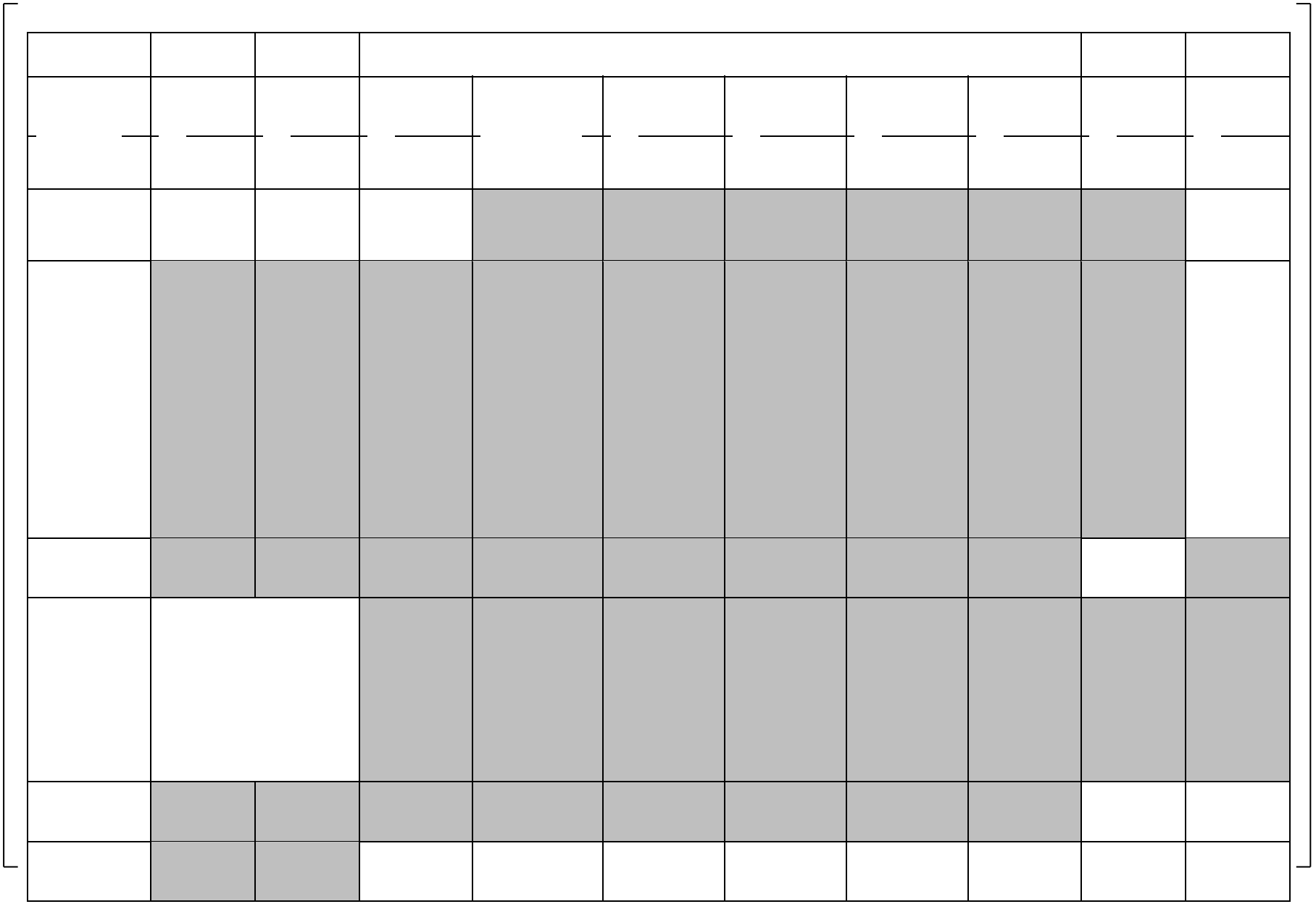
a,c



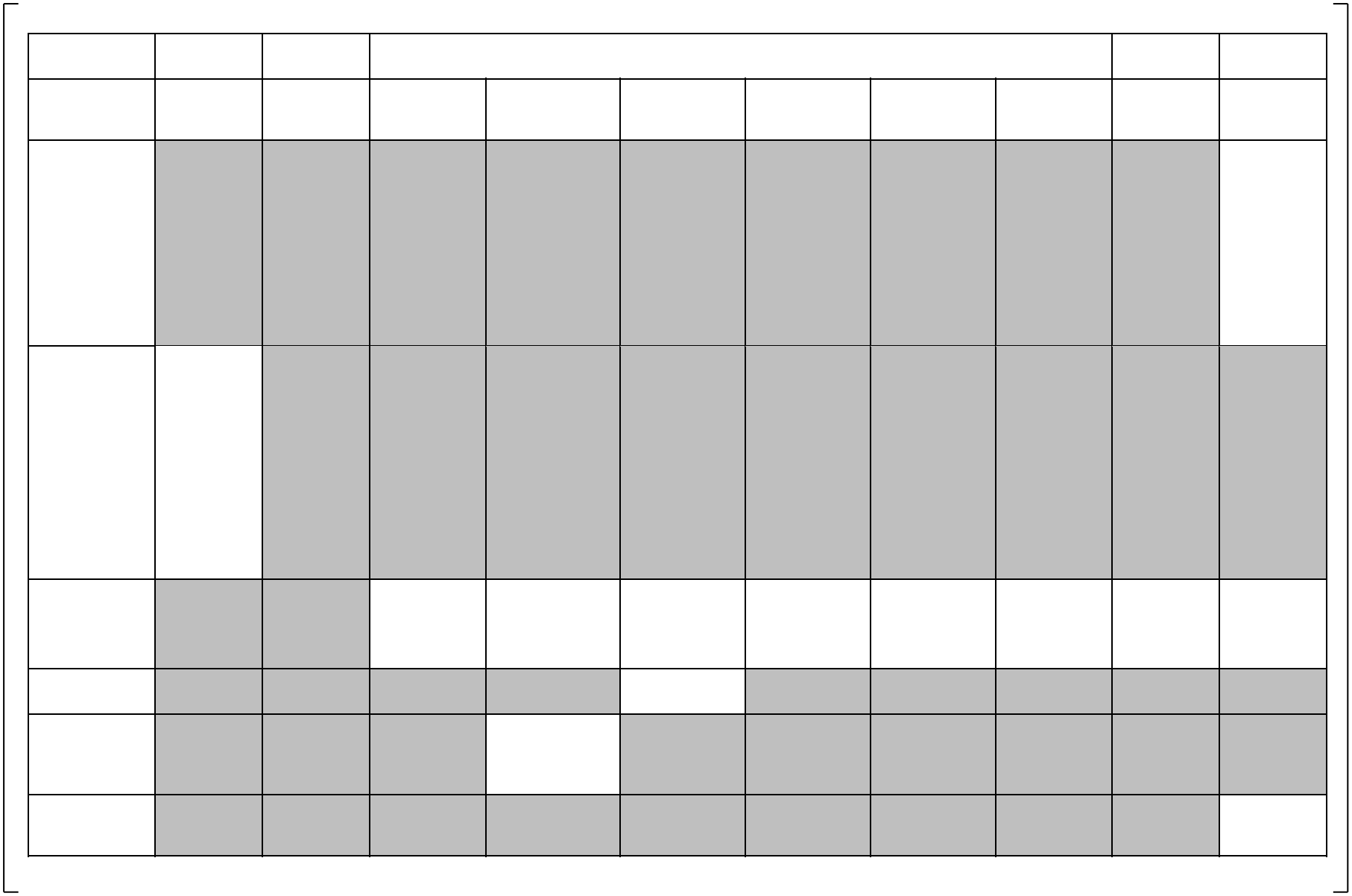
a,c



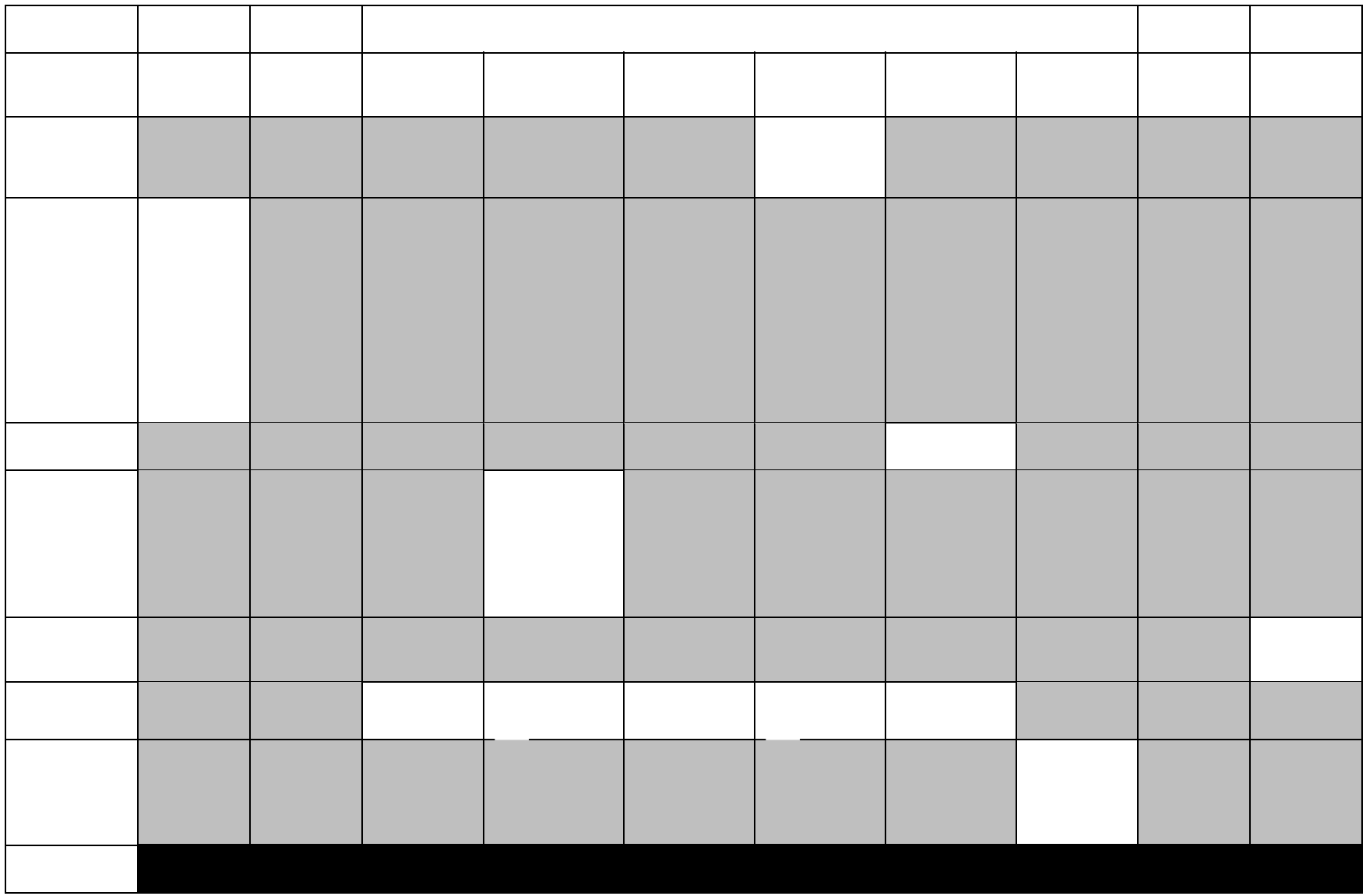
a,c



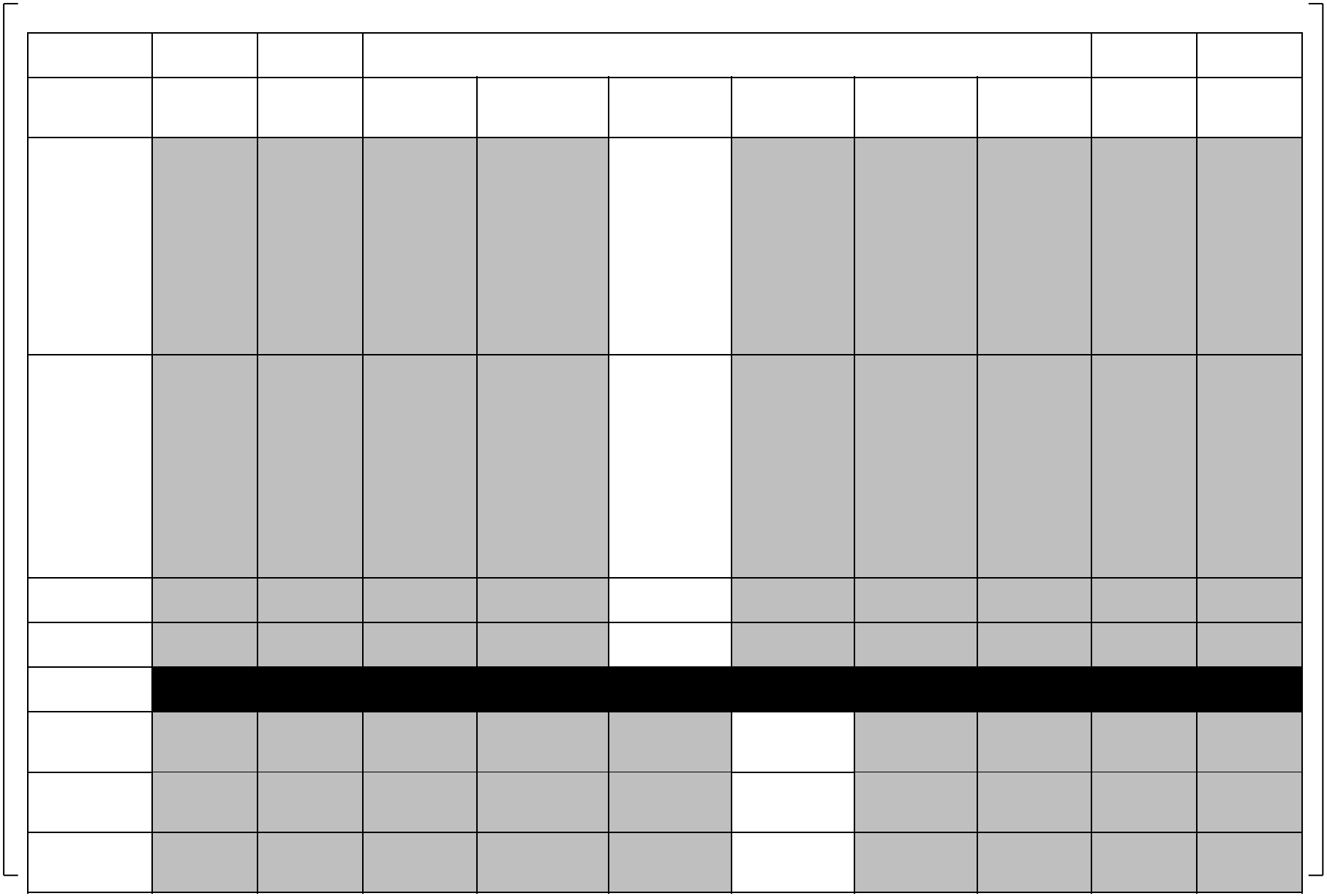
a,c



a,c



a,c



a,c

EXHIBIT 6-1 SOFTWARE CHANGE REQUEST FORM

SCR # _____

Date: _____ CUSTOMER: _____ Page 1 of ___

Subject: _____ Software Affected:_____ Originator: _____ Version: _____ Revision: _____ Classification[ ]: 1-Emergency 2-Corrective 3-Adaptive 4-Perfective
Summary of Requested Change:
Reason for Change:
Documents Affected (Document No./Revision):
_____ Design Approval/Date: _____ Engineering Project Manager/Date

Implementation Completed: (Including Documentation) _____ Implementation Engineer/Date	Testing Completed: Exception Report #: _____ Documentation: _____ Review/Date: _____

EXHIBIT 7-1 COMPARISON OF SYSTEM VALIDATION TEST AND FAT

Test Item	Design Aspect	System Validation Test (First Application)	FAT (Nth Application)
Software integration	Software integration	Application software loads into target hardware; CRC check confirms no memory errors; capacity and cycle time checks consistent with design documentation	Same as for first application
Safety functions	Safety functions	Each safety function (reactor trip and engineered safeguard features [ESF]) shown to properly respond to each input per functional logic diagrams (FLDs); all component actuation outputs respond to system-level actuations appropriately; manual actuations at system and component level are effective	Each safety function (reactor trip and ESF system actuations) shown to properly respond to each initiating input; component actuation outputs respond to system-level actuations appropriately; manual actuations at system and component level are effective. (Demonstrates trips and actuations are functioning but does not need to retest the software logic that has been previously verified in the first application. Time response testing can be used to demonstrate trips and actuations).
	Voting logic	All combinations including bypasses and forced trips	Subset of combinations demonstrating that each input to voting logic is effective. (Time response testing can be used to demonstrate voting logic inputs from each division.)
Communications	Intra-cabinet communications	Each signal shown to connect to every intended destination	Links confirmed to be operational through diagnostics; no signal tracing

Test Item	Design Aspect	System Validation Test (First Application)	FAT (Nth Application)
	Inter-cabinet communications (within channel and between channels)	Each signal shown to connect to every intended destination	Links confirmed to be operational through diagnostics; representative signals are traced
Displays	Display navigation	All designed displays loaded and accessible through various navigation means	All designed displays loaded.
	Signal value display	Each display shows values correctly formatted over signal range including display of abnormal conditions; trend functions demonstrated	Single value for representative sample of signals is displayed (background displayed and foreground operating with real data)
	Soft operator controls	All soft controls demonstrated to be effective, including operator dialog sequences, and test sequences	Sampling of soft controls for plant operations (not maintenance) demonstrated to be effective per display
Diagnostics	System health diagnostics	Abnormal conditions simulated to demonstrate correct operation of status signals and alarms	No unexpected off normal conditions created (health displays used to confirm normal system status)
	Error handling	Random hardware failures; for example, single sensor, single power supply. Errors shall be handled with known consequences.	Hardware operability such as sensor inputs checks.

Test Item	Design Aspect	System Validation Test (First Application)	FAT (Nth Application)
Performance	Software functionality of other functions	Comprehensive logic and functional algorithm testing at the system level; testing shows connection of each input and output signal to function algorithm	Tested only as it relates to operability of the hardware. This testing to be determined by V&V organization based on the need for the test to demonstrate variability that is possible from the assembly or manufacturing of the hardware. Examples may include hardware interlocks, hardware setpoints that have software interfaces, or functionality that is dependent on hardware configuration
	Signal redundancy	Shared inputs produce the same output from redundant processors	Sampling of the redundancy to the extent that indicates that the redundancy is effective in selection
	I/O connectivity	Testing shows connection of each input and output signal to function algorithm	Confirmed as part of safety functional and response time tests and in the hardware tests in combination of V&V testing of software and system

Test Item	Design Aspect	System Validation Test (First Application)	FAT (Nth Application)
	Time response testing	Multiple runs of representative trip and ESF functions to validate analytic modeling and confirm compliance to requirement	A representative sample of safety function tests on the deliverable hardware with the deliverable software to demonstrate critical safety trips, consistency with analytic model and first application response tests • One path through critical hardware component; e.g., each PM, I/O module, high-speed datalink, etc. • Component response confirmed by commercial grade dedication process (similar to spare parts).
	Abnormal communications events	Loss of link conditions simulated and shown to be handled correctly	Links confirmed to be operational through diagnostics; no signal tracing
	Loss of power and restoration	Demonstrate expected behavior of system outputs on loss of power; proper initialization on restoration of power	Sampling to demonstrate no spurious activity due to full division loss of power and restoration
	Function independence	Demonstrate that no adverse interactions exist between independent functions	Not performed in Nth application FAT

EXHIBIT 10-1 COMPUTER CODE CERTIFICATE

The following computer code, as noted by its name, version number and executable file identification, is approved for design use.

System Name: _____

Code Name: _____

Version/Revision Number: _____

Executable File Identification: _____

Computer(s): _____

Restrictions (List any limitations on use, special hardware considerations, etc.): _____

_____

_____

Listed are the software modules and their current revision (use additional pages as necessary):

Module Name/Classification	Version/Revision

Verification and Validation Report Number: _____

IV&V Team Leader: _____ Date: _____

EXHIBIT 11-1 EXCEPTION REPORT

Exception Report Number _____

System Name:	Plant:
Procedure Name:	Procedure Number:
Tester Name:	Rev.:
Summary of Exception: Class:	Date: Step:
Resolution: Responsibility:	
Implementation: ☐ Procedure Correction ☐ Software Change Implemented By: Retested By: Reviewed By:	 Date: Date: Date:

(Last Page of Section 13)