Westinghouse

Westinghouse Electric Company
Nuclear Services
P.O. Box 355
Pittsburgh, Pennsylvania 15230-0355
USA

Document Control Desk
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001

Direct tel: 412-374-4643
e-mail: greshaja@westinghouse.com
Our ref: LTR-NRC-05-4

January 21, 2005

Project Number 700

Subject: Transmittal of Westinghouse Topical Report WCAP-16096-NP-A, Revision 1A, "Software Program Manual for Common Q Systems," dated December 2004.

Reference: Letter, H. Berkow (NRC) to J. Gresham (Westinghouse), "Final Safety Evaluation for Topical Report WCAP-16096-NP-A, Revision 1, 'Software Program Manual for Common Q Systems' (TAC No. MC2294)," dated September 28, 2004.

On September 28, 2004, the Nuclear Regulatory Commission issued the final safety evaluation for topical report WCAP-16096-NP-A, Revision 1, "Software Program Manual for Common Q Systems." As requested by the Reference letter, Westinghouse has compiled an accepted version of this report which is designated as Revision 1A and dated December 2004.

The purpose of this letter is to transmit the accepted version of topical report WCAP-16096-NP-A, Revision 1A, for NRC files. A total of four copies of this report, one copy for the NRC Document Control Desk and three copies for the project manager, are transmitted herewith pursuant to the copy requirements specified on the NRC topical report web page.

Westinghouse has determined that WCAP-16096-NP-A, Revision 1A is nonproprietary and may be placed in the public domain.

Please contact Mark Stofko at 724-733-6780 if you have any questions.

Very truly yours,

JA Gresham

James A. Gresham, Manager
Regulatory Compliance & Plant Licensing

cc: W. A. Macon (NRC) 3 copies
B. Benney (NRC)
L. Feizollahi (NRC)

Enclosures:

D054

Westinghouse Non-Proprietary Class 3

WCAP-16096-NP-A
Revision 1A

December 2004

Software Program Manual for Common Q Systems

Westinghouse

Westinghouse

Westinghouse Non-Proprietary Class 3

Software Program Manual for Common Q Systems

WCAP-16096-NP-A

Revision 1A

December 2004

APPROVALS

Official Record Electronically Approved in EDMS 2000

Function	Name and Signature
Author	Mark J. Stofko Principal Engineer, Regulatory Compliance & Plant Licensing
Reviewed	Kevin R. Finlay Principal Engineer, Design Basis Engineering
	Kenneth A. Kloes Engineer, Quality Programs and Supplier Qualification
Approved	Kenneth G. Lunz Product Manager, Reactor Protection Systems

WESTINGHOUSE NON-PROPRIETARY CLASS 3

© 2004 Westinghouse Electric Company LLC
All Rights Reserved

LIST OF CONTRIBUTORS/REVIEWERS

Name and Title	Date
Joseph E. Burns, Software Engineer	January 2004
Norbert N. Carte, V&V Engineer	January 2004
Warren R. Odess-Gillett, Software Engineer	January 2004
Frank J. Safryn, V&V Engineer	January 2004
Stephen Slinski, Software Engineer	January 2004

This Page Intentionally Left Blank

NRC SAFETY EVALUATION AND COVER LETTER – SEPTEMBER 28, 2004

This section contains the following documents:

- NRC cover letter, "Final Safety Evaluation for Topical Report WCAP-16096-NP-A, Revision 1, "Software Program Manual for Common Q Systems" (TAC No. MC2294)

- "Safety Evaluation by the Office of Nuclear Reactor Regulation, Topical Report WCAP-16096-NP-A, Revision 1, 'Software Program Manual for Common Q Systems', Westinghouse Electric Company, Project No. 700"

September 28, 2004

Mr. James A. Gresham, Manager
Regulatory and Licensing Engineering
Westinghouse Electric Company
P.O. Box 355
Pittsburgh, PA 15230-0355

SUBJECT: FINAL SAFETY EVALUATION FOR TOPICAL REPORT WCAP-16096-NP-A, REVISION 1, "SOFTWARE PROGRAM MANUAL FOR COMMON Q SYSTEMS" (TAC NO. MC2294)

Dear Mr. Gresham:

On January 29, 2004, Westinghouse Electric Company (Westinghouse) submitted Topical Report (TR) WCAP-16096-NP-A, Revision 1, "Software Program Manual for Common Q Systems," to the staff. On September 3, 2004, an NRC draft safety evaluation (SE) regarding our approval of the TR was provided for your review and comments. By e-mail dated September 22, 2004, Westinghouse stated that they had no comments on the draft SE.

The staff has found that WCAP-16069-NP-A, Revision 1, is acceptable for referencing in licensing applications to the extent specified and under the limitations delineated in the TR and in the enclosed SE. The SE defines the basis for 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 non-proprietary version of this TR within three months of receipt of this letter. The accepted version shall incorporate this letter and the enclosed SE between the title page and the abstract. It must be well indexed such that information is readily located. Also, it must contain historical review information, such as questions and accepted responses, draft SE comments, and original TR pages that were replaced. The accepted version shall include a "-A" (designating accepted) following the TR identification symbol.

If future changes to the NRC's regulatory requirements affect the acceptability of this TR, Westinghouse and/or licensees referencing it will be expected to revise the TR appropriately, or justify its continued applicability for subsequent referencing.

Sincerely,

/RA/

Herbert N. Berkow, Director
Project Directorate IV
Division of Licensing Project Management
Office of Nuclear Reactor Regulation

Project No. 700

Enclosure: Safety Evaluation

cc w/encl:
Mr. Gordon Bischoff, Manager
Owners Group Program Management Office
Westinghouse Electric Company
P.O. Box 355
Pittsburgh, PA 15230-0355

If future changes to the NRC's regulatory requirements affect the acceptability of this TR, Westinghouse and/or licensees referencing it will be expected to revise the TR appropriately, or justify its continued applicability for subsequent referencing.

Sincerely,
/RA/
Herbert N. Berkow, Director
Project Directorate IV
Division of Licensing Project Management
Office of Nuclear Reactor Regulation

Project No. 700

Enclosure: Safety Evaluation

cc w/encl:
Mr. Gordon Bischoff, Manager
Owners Group Program Management Office
Westinghouse Electric Company
P.O. Box 355
Pittsburgh, PA 15230-0355

DISTRIBUTION:
PUBLIC
PDIV-2 Reading
RidsNrrDlpmLpdiv (HBerkow)
RidsNrrDlpmLpdiv2 (RGramm)
RidsNrrWMacon
RidsNrrLAEPeyton
RidsOgcRp
RidsAcrsAcnwMailCenter
EMarinos

ADAMS Accession No.: ML042730580 NRR-106

OFFICE	PDIV-2/PM	PDIV-2/LA	EEIB-A/SC*	PDIV-2/SC	PDIV/D
NAME	WMacon	EPeyton	EMarinos	RGramm	HBerkow
DATE	9/22/04	9/27/04	8/25/04	9/28/04	9/28/04

OFFICIAL RECORD COPY

*** SE dated 8/24/04.**

SAFETY EVALUATION BY THE OFFICE OF NUCLEAR REACTOR REGULATION

TOPICAL REPORT WCAP-16096-NP-A, REVISION 1

"SOFTWARE PROGRAM MANUAL FOR COMMON Q SYSTEMS"

WESTINGHOUSE ELECTRIC COMPANY

PROJECT NO. 700

1.0 INTRODUCTION

By letter dated January 29, 2004 (ADAMS Accession Number ML040360115), Westinghouse Electric Company (Westinghouse) submitted Topical Report (TR) WCAP-16096-NP-A, Revision 1, "Software Program Manual for Common Q Systems," to the NRC for review.

Two previous versions of this Software Program Manual (SPM) have been submitted and reviewed. The original version was submitted by CE Nuclear Power on June 5, 2000 (ADAMS Accession Number ML003722925), as a part of CENPD-396-P, Revision 1, "Common Qualified Platform." This TR and the SPM were approved in a safety evaluation (SE) dated August 11, 2000 (ADAMS Accession Number ML003740165). When Westinghouse acquired the Common Qualified (Common Q) platform product line, an updated version of the SPM was submitted on August 14, 2002 (ADAMS Accession Number ML003721618), as WCAP-16096-NP-A, Revision 0. This TR was approved in a SE dated February 24, 2003 (ADAMS Accession Number ML030550776).

WCAP-16096-NP-A, Revision 1 specifies plans for implementing a structured software life cycle process for application software and provides guidance for configuration management of commercial-grade hardware and previously-developed software.

2.0 REGULATORY EVALUATION

The basic review criteria for the evaluation of the changes to the SPM is that the changes do not compromise the characteristics of the Common Q platform that were critical to its acceptability in the initial SE. Therefore, the review criteria for the changes are essentially all of the criteria that applied to the previously approved the Common Q platform. These criteria are identified in Section 3.1, "Review Criteria," in the initial SE dated August 11, 2000. Since this SPM has been previously approved, this SE will only evaluate the changes to ensure that they continue to meet these criteria. For this review, the staff has determined that the following subset of regulatory guides (RG) and standards applies to the acceptability of the changes:

- RG 1.152-1996, "Criteria for Digital Computers in Safety Systems of Nuclear Power Plants" (which endorses the Institute of Electrical and Electronic Engineers (IEEE) Standard (Std) 7-4.3.2, "IEEE Standard for Digital Computers in Safety Systems of Nuclear Power Generating Stations").

- RG 1.168-1997, "Verification, Validation, Reviews, and Audits for Digital Computer Software Used in Safety Systems of Nuclear Power Plants" (which endorses the American National Standards Institute (ANSI)/IEEE Std 1012, "IEEE Standard for Software Verification and Validation Plans," and IEEE Std 1028, "IEEE Standard for Software Reviews and Audits").

- RG 1.170-1997, "Software Test Documentation for Digital Computer Software Used in Safety Systems of Nuclear Power Plants" (which endorses IEEE Std 829, "IEEE Standard for Software Test Documentation").

- RG 1.171-1997, "Software Unit Testing for Digital Computer Software Used in Safety Systems of Nuclear Power Plants" (which endorses ANSI/IEEE Std 1008, "IEEE Standard for Software Unit Testing").

- RG 1.173-1997, "Developing Software Life Cycle Processes for Digital Computer Software Used in Safety Systems of Nuclear Power Plants" (which endorses IEEE Std 1074, "IEEE Standard for Developing Software Life Cycle Processes").

3.0 TECHNICAL EVALUATION

The Westinghouse SPM for Common Q systems specifies plans for implementing a structured software life cycle process for application software and provides guidance for configuration management of commercial-grade hardware and previously developed software. As stated in the initial SE, licensees using the Common Q platform for plant-specific applications are required to implement the application software in accordance with the SPM. As stated above in this SE, the staff has already reviewed the SPM twice before; therefore, this SE will only evaluate the changes to ensure that they continue to meet the applicable review criteria.

Westinghouse has stated that the SPM was modified for the following reasons:

- To ensure consistency with updated Westinghouse Repair, Replacement and Automation Services (RRAS) quality program requirements and to meet the RRAS documentation standards.

- Experience with the Palo Verde Core Protection Calculator System (CPCS).

- Implemented changes based on NRC review and audit experience.

- Corrected errors and inconsistencies discovered during early experience with Common Q projects.

- To correct typographical errors and to clarify some of the process descriptions.

3.1 Changes Based on RRAS Program Requirements

These changes are primarily format changes. The acronym section, definitions section, and references section were moved to different locations. References to the ABB Quality Assurance Manual were removed, and some definitions were added or modified. The title of

"V&V Team Leader" was replaced with "V&V Engineering Line Manager;" however, the administrative and financially independence from the design team manager was not changed. The other change was to show that the Quality organization is responsible for auditing the software safety plan implementation and performing process certification rather than the verification and validation (V&V) team and that the results of the audit are documented in the Quality organization's Audit Report. These changes do not compromise the characteristics that were critical in the initial SE to meet the review criteria, and are, therefore, acceptable.

3.2 Changes Based on Palo Verde CPCS Experience

These changes are to provide clarification rather than to make substantive changes. The first change is to provide clarification that the independent managerial review to assess the execution of all of the actions and the items identified in the Software Quality Assurance Plan (SQAP) that is included in the SPM, as required by IEEE Std 730, is the responsibility of the Quality organization. The second change is to provide clarification that the requirements for the hardware design process are defined in the Westinghouse Policy and Procedures Manual and that hardware verification is performed as part of the hardware quality assurance activities, which is also defined in the Westinghouse Policy and Procedures Manual.

These changes do not compromise the characteristics that were critical in the initial SE to meet the review criteria, and are, therefore, acceptable.

3.3 Changes Based on NRC Review and Audit Experience

These changes provide clarification of items based on comments and observations during staff reviews, rather than to make substantive changes.

These changes provide a clearer definition of responsible groups and organizational structure, such as the responsibilities of the V&V Group, the Software Librarian and the Automation Engineering groups. In addition, the Requirements Traceability Analysis, Requirements Traceability Matrix, and the documentation requirements are better defined.

These changes do not compromise the characteristics that were critical in the initial SE to meet the review criteria, and are, therefore, acceptable.

3.4 Changes to Implemented Design Process Improvements and Corrected Errors and Inconsistencies Based on Early Experience with Common Q Projects

There are a number of minor changes to the SPM which further clarify or modify the software processes. Some of these changes are as follows:

- Define when software will be entered into a controlled access account.
- State that the results of physical reviews shall be documented with a Certificate of Conformance.
- Add a statement that non-commercial software (freeware) cannot be used for Protection Class Software.

- Add a statement that the coding standards to be applied to a project shall be referenced in the Project Quality Plan, and that the V&V team shall review the applicable coding standards for each project for acceptability.

- Consistently use "SDD" to mean Software Design Description rather than Software Design Document and "SCR" to mean Software Change Request rather than System Change Request.

- Require that Westinghouse personnel assigned to work on any activity in the software life cycle process must complete training on the SPM in accordance with the "NA Policy and Procedures Manual," and that the V&V team must review training materials prepared for the customer.

- Revise the statement so that in-process audits shall be performed by the Quality organization rather than the V&V team, and that these audits shall be documented in an audit report rather than the V&V report.

- Require that the Software quality assurance plan be required for all quality classifications defined for the Common Q system: protection, important-to-safety, important-to-availability, and general purpose software, not just software "within the scope of Westinghouse software."

- Revise the responsibilities within the SPM so that:

    1. Verification of the implementation of quality assurance requirements is performed by the Quality organization, and that the V&V Team Leader shall verify that software and associated documentation has been developed in accordance with the standards specified in the SQAP.

    2. The responsibility for performing the Physical Review is moved from the V&V team to the design team.

    3. Stating that the Preliminary Hazards Analysis Report can be either completed by the V&V team or completed by the design team and reviewed by the V&V team.

There are a number of other minor changes to terminology, documentation requirements and processes, Westinghouse standard forms, and dates on references and standards. The staff has reviewed these changes, and has determined that these changes are minor, and do not alter the SPM in a significant manner.

These changes do not compromise the characteristics that were critical in the initial SE to meet the review criteria, and are, therefore, acceptable.

3.5 Changes to Correct Typographical Errors and to Clarify Some of the Process Descriptions

These changes are being made to correct some errors in the original document, and to provide clarification of some process descriptions. The staff has reviewed these changes, and has

determined that these changes are minor, and do not alter the SPM in a significant manner. Since these changes do not compromise the characteristics that were critical in the initial SE to meet the review criteria, they are acceptable.

4.0 CONCLUSION

On the basis of this review, the staff has determined that the proposed changes to the Westinghouse SPM for Common Q systems continue to meet the applicable review criteria and do not compromise any of the characteristics that were critical in the initial SE. Therefore, the changes in WCAP-16096-NP-A, Revision 1, are acceptable.

Principal Contributor: P. Loeser, NRR

Date: September 28, 2004

This Page Intentionally Left Blank

REVISION HISTORY

RECORD OF CHANGES

Revision	Revision Made By	Description	Date
0	Warren R. Odess-Gillett	Original Issue of WCAP document	May 2003
1	Mark J. Stofko	Revised to be consistent with current QA procedures and to implement process improvements.	January 2004
1A	Mark J. Stofko	Revised to issue the NRC accepted version.	December 2004

File Name: WCAP-16096-NP-A r1A

File Size: 3105280

This Page Intentionally Left Blank

DETAILED RECORD OF CHANGES

Revision 0 **May 2003**

Description:
This is the original issue of this document. This document was previously released as CE-CES-195. It is being prepared to create the accepted version in accordance with the USNRC Safety Evaluation dated February 24, 2003. The previously released document CE-CES-195, Revision 1 has been modified as follows:

- A document titled, "Changes to the Westinghouse Software Program Manual for Common Q Systems, CE-CES-195, Revision 1" was provided to the NRC via LTR-NRC-02-41, dated August 14, 2002. The changes identified in that document have been incorporated into this revision.
- Minor changes were made to clarify some of the process descriptions.
- Correction of typographical errors.
- Minor document format changes were made.
- In accordance with established NRC procedures, the accepted version of the Common Q Topical Report (WCAP-16097-P-A) includes the original Safety Evaluation, two supplemental Safety Evaluations and associated cover letters that were issued by the NRC. This Software Program Manual was treated as an appendix to the Common Q Topical Report during the NRC review and approval process.
- References to the ABB Quality Assurance Manual and procedures were replaced with the Westinghouse Quality Management System and its implementing procedures. Westinghouse letter I&CE-LIC(02)-114 dated 11/14/02 provides a mapping to the applicable implementing procedures. Changing the references to the Westinghouse Quality Management System and its implementing procedures does not in any way change the specific life cycle processes described in the SPM.

Revision 1 **January 2004**

Description:
This document provides the software design methodology for the design and development of nuclear safety systems. This document has been revised to ensure consistency with updated Repair, Replacement and Automation Services (RRAS) quality program requirements. The document has also been reformatted to be consistent with RRAS documentation standards. In addition, design process improvements have

been implemented based on early experience with Common Q projects. The following is a list of specific changes:

- Implemented format changes to be consistent with RRAS documentation standards.
  - — Moved acronyms from Section 1.5 to "Acronyms and Trademarks" beginning on page xxix.
  - — Moved definitions from Section 1.5 to "Glossary of Terms" beginning on page xxxi.
  - — Moved References from Section 1.6 to "References" beginning on page xxxiii.
  - — Moved acronyms in Section 6.1.3 to the Acronyms and Trademarks section. Added SysRS and clarified use of SysRS and SRS.
  - — Moved definition of "software item" from Section 6.1.3 to the Glossary of Terms section.
- Implemented changes based on Palo Verde CPCS experience.
  - — Addressed APS audit finding (reference: CAPS Issue 02-353-W008 and Commitment 02-339-W007.02). This SPM defines the processes associated with software development including the software/hardware interface. This change provided clarification that the requirements for the Common Q hardware design process are defined in the Westinghouse Policy and Procedures Manual and that hardware verification is performed as part of the hardware quality assurance activities that are also defined in the Westinghouse Policy and Procedures Manual. Removed specific V&V activities for hardware since the design team performs these activities. Provided additional clarification regarding V&V activities for the hardware/software interface. This change affected sections 1.1, 5.1, 5.4.4.1, 5.4.5.2, 5.5.3.2, 5.5.4.2, and 5.5.7.2.
  - — Addressed APS audit finding (reference: CAPS Issue 02-353-W006 and Commitment 02-339-W007.02). This change provided clarification that the independent managerial review required by IEEE Std 730 is the responsibility of the Quality organization. The purpose of the review is to assess the execution of all of the actions and the items identified in the SQAP that is included in the SPM. This change affected section 4.6.2.8.
- Westinghouse letter I&CE-LIC(02)-114 dated 11/14/02 provided a mapping to the applicable Westinghouse Quality Management System and its implementing procedures. Modifications were made to align the SPM processes to the RRAS quality program requirements so that the interim-mapping document would no longer be required. This change affected all sections of the document as follows:

— Removed references to the ABB Quality Assurance Manual in EXHIBIT 1-1 RELATIONSHIP OF SPM TO IEEE STANDARDS (previously titled SOFTWARE PROGRAM DOCUMENTATION HIERARCHY).

— Replaced references to the Cognizant Engineering Organization (CEO), Cognizant Engineer and CEO Manager with organizational terms and responsibilities consistent with the RRAS Operational Model throughout the document. Definitions of Automation Engineering, ELM, EPM and Platform Lead were added to the Glossary of Terms on page xxxi. Revised the resource summary descriptions in Section 5.4.3 to be consistent. Also, revised EXHIBIT 2-1 DESIGN/V&V TEAM ORGANIZATION to use consistent terminology. Provided additional clarification in SECTION 2 and in Section 5.4.3.2 that the V&V team reports to an Engineering Line Manager (ELM) who is administratively and financially independent from the design team manager. Replaced "V&V Team Leader" with "V&V ELM" for the responsibilities described in Section 3.3.1.

— Added definition of SAP to the Glossary of Terms on page xxxi. Also, referenced SAP in Section 3.3.2.

— Removed "safety related software" and "Quality Class 1" terminology and related the software classification scheme to the NA Policy and Procedures Manual. Added requirement to complete a NuCAR for software items not classified in EXHIBIT 4-1 ASSIGNMENT OF COMMON Q SOFTWARE TO CLASSES.

— Added definition of the Quality organization to the Glossary of Terms on page xxxi and to Section 3.3.1. Revised Section 4.6.2.7 to state that the Quality organization shall perform in-process audits for Common Q systems including all software classes. Also, replaced "NQA" with "Quality" in Section 8.5.2.

— References to "project plan" were replaced with "Project Quality Plan".

— Deleted Reference 29, "ABB CENO Proposal Development and Project Execution Process Manual". The document is obsolete.

— Clarified SECTION 2 by replacing the sentence, "Verification of the implementation of quality assurance requirements . . ." with "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."

— Revised Sections 3.3.3 and 3.3.6 to indicate that the Quality organization is responsible for auditing the software safety plan implementation and performing process certification rather

than the V&V team. Also, the results of the audit are documented in the Quality organization's Audit Report.

— Revised Section 4.5.1 to state that "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."

- Implemented design process improvements based on early experience with Common Q projects.

  — Added statement to Section 1.1 that additions, deletions and alternatives to the SPM requirements in order to be compliant with IEC 880 are defined in a separate document.

  — Revised Section 4.6, Reviews, to more accurately state how review requirements are achieved and where they are documented.

  — Revised Section 6.2.2.3 to state that 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.

  — The Test Exception Report (TER) form is used to document all software anomalies, not just test exceptions. Therefore, the name was changed to Exception Report. All occurrences of "TER" and "Test Exception Report" were replaced with "Exception Report".

  — Revised Common Q Comment Record form in SECTION 10 (formerly Exhibit 4-5) to add document revision number and to delete signature requirements.

  — Deleted the Controlled Software Distribution List form in SECTION 10 (formerly Exhibit 6-2). Software Release Records provide the details for software distributions. This change also affected Sections 6.2.2.6, 6.2.2.7 and 6.3.3.

  — Deleted the Software Problem Report form in SECTION 10 (formerly Exhibit 6-3). Westinghouse will accept a software problem notification from a customer in any form. For internal software problems, Exception Reports are used. This change also affected Sections 4.5.2.4, 6.1.1, 6.2.2.5, 6.2.2.6, 6.3.2, 6.3.3, 9.3 and 9.4.

  — Deleted the Computer Code Error Notification form in SECTION 10 (formerly Exhibit 7-1). Customers will be notified of software errors by issuing "Technical Bulletins" in accordance with NA Policy and Procedures (Reference 4). No special form is required. This change also affected Sections 6.2.2.6 and 9.3.

— Deleted the Computer Code Error Notification Response Receipt form in SECTION 10 (formerly Exhibit 7-2). 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 that Westinghouse issues for error notification. This change also affected Section 9.3.

— Revised Section 6.3.2 to state that the "Software Change Request Procedure" shall be implemented via an automated process.

— Revised Section 9.1 to state that the exception reporting procedure shall be implemented via an automated process.

— Revised Section 4.6.2.6 to state that the results of physical reviews shall be documented with a Certificate of Conformance in accordance with the requirements of Reference 4.

— Revised Section 4.1.2 to add additional requirements for the use of "freeware". This software can not be used for Protection Class software. Also, it will be subject to V&V activities starting with the implementation phase. If the software has poor, or no, documentation; then documentation must be prepared.

— Added statement to the Post Mortem Review, Section 4.6.2.10, as follows, "Project metrics should be reviewed at this time to determine if any process improvements can be identified."

— CAPS Issue 03-169-W001 identified a suggestion for improvement related to the documentation of application specific coding standards and guidelines. Revised Section 4.5.2.1 to state that coding standards to be applied to a project shall be referenced in the Project Quality Plan, and that the V&V team shall review the applicable coding standards for each project for acceptability. Also, added a line item to EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST for the reviewer to verify conformance with coding standards and guidelines.

— Updated references to the latest IEEE standards revision, up to and including 1998. In some cases the current regulatory guides related to software, reference earlier revisions of the standards. However, none of the original requirements that were previously endorsed has been removed from the later revision. This change affected the Reference list on page xxxiii.

- Corrected errors and inconsistencies discovered during early experience with Common Q projects.

— Consistently use "SDD" to mean Software Design Description (per IEEE Std 1012) rather than Software Design Document in Sections 5.5.4.1 and 8.3.

— Consistently use "SCR" to mean Software Change Request rather than System Change Request in Section 6.3.2.

— Deleted the embedded reference to IEEE Std 729-1983 in Section 6.1.3 and added a pointer to IEEE Std 610.12-1990 in the Glossary of Terms on page xxxi. IEEE Std 610.12-1990 was already in the Reference list. This is an update and expansion to IEEE Std 729-1983.

— Deleted Reference 25 (IEEE/EIA Standard 12207.1) and the pointer in Section 5.1. The Common Q software life cycle process was developed consistent with IEEE Std 1074 (Reference 24).

— Added explicit reference to IEEE Std 1028 in Section 4.6.1 regarding software reviews and audits.

— Revised Section 3.5.1 to indicate that Westinghouse personnel assigned to work on any activity in the software life cycle process must complete training on the SPM in accordance with the NA Policy and Procedures Manual. Also, the V&V team must review training materials prepared for the customer.

— Revised Section 3.6 to indicate that management approval of the Software Safety Plan is indicated by signature on the SPM cover sheet.

— Provided clarification on how the results of software safety analysis tasks are documented. This change affected Sections 3.3.5.11, 3.3.5.12 and 3.3.6.

— Provided clarification in Section 5.5.6 that the Software Safety Test Analysis is documented by completing EXHIBIT 5-5 CHECKLIST NO. 4, SOFTWARE VERIFICATION AND VALIDATION TEST PHASE CHECKLIST. Added a line item to the Checklist with respect to software hazard prevention.

— Provided clarification and additional details in Sections 5.5.8 and 5.7.2 regarding the documentation of V&V activities including regression analysis for system modifications.

— Provided clarification in Section 8.5.1 that software items to be evaluated and custom software to be developed can be identified in a project specific SVVP as an alternative to the Project Quality Plan (PQP). If a project specific SVVP is written, then it must be referenced in the PQP.

— Provided clarification in Section 8.6 that user documents can reference (instead of "consist of") vendor documents. Also, stated that user documents can be combined into a single Technical Manual.

— Provided clarification in Section 8.7 that project specific SCMP details can be included in a project specific SCMP as an alternative to the Project Quality Plan (PQP). If a project specific SCMP is written, then it must be referenced in the PQP.

— Revised EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES as follows:

- Changed "Module Test Execution" to "Module Test Execution / Report".
- Changed "Unit Test Execution" to "Unit Test Execution / Report".
- Changed "SAT" line items to "Installation Test" and added footnote to these items to state, "Applicable if in Westinghouse scope of supply."
- Changed responsibility from "VT/DT" to "VT/DT*" for Test Plan row / Protection and Important-to-Safety columns.
- Changed responsibility from "DT" to "DT/VT" for Module Coding row / Protection and Important-to-Safety columns.
- Changed responsibility from "DT/VT" to "VT/DT*" for the Module Test Procedure row and Module Test Execution row / Protection column.
- Changed responsibilities for Unit Testing to be consistent with Module Testing.

— Removed statement, "Items identified with an asterisk (*) require that the reviewer attach his notes . . ." from Checklists 1 through 5.

— Provided clarification in Sections 4.3.2, 4.3.2.4, 5.5.5.1, 5.5.6.1 and 8.4 that source code listing documents are not prepared. Source code documentation includes software release records and code review reports.

— Revised EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST to add a check of the compliance with WCAP-7211, Revision 4, with respect to "Control of Computer Programs" (Reference 31) and coding standards and guidelines. Reference 31 was a new addition to the list of references.

— Added explicit reference to EXHIBIT 9-2 CORRECTIVE ACTIONS PROCESS in Section 9.4.

— Revised Section 6.3.4 to state that In-process audits shall be performed by the Quality organization rather than the V&V team to verify the consistency of the design process and for

proper implementation of the software QA process. Also these audits shall be documented in an audit report rather than the V&V report.

— Add "System Name _____" to EXHIBIT 8-1 COMPUTER CODE CERTIFICATE.

— In Section 5.1.3, changed "The generic documentation plan defines the structure and format of contents of the documents which may be produced during various phases of the project." to "SECTION 8 defines the structure and format of the documents that may be produced during various phases of the project".

— In EXHIBIT 4-3 TASKS REQUIRED FOR SOFTWARE CATEGORIES (formerly Exhibit 4-4) changed "Inter-System Test" to "Integration Test". Also, Deleted the sentence, "Inter-system testing (the process of testing system interfaces) shall be performed in accordance with the Test Plan" in Section 4.3.2.5. System interfaces are tested during System Testing.

— Added the definitions of *Module*, *Unit* and *System* to the Glossary of Terms on page xxix.

— Removed the phrase "within the scope of Westinghouse software" in Section 4.1.2. The sentence that describes the scope of the SQAP now reads as follows: "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."

— Revised Section 4.3.1 to clarify that the EPM shall ensure that all design team activities are in accordance with the SQAP and that verification of the implementation of quality assurance requirements is performed by the Quality organization. The V&V Team Leader shall verify that software and associated documentation has been developed in accordance with the standards specified in the SQAP. This includes ensuring that the coding standards (Section 4.5.2.1), testing standards established in the test plan and documentation standards (SECTION 8) have been followed.

— Added a clarification in Section 4.3.2.2. Changed a topic included in the test plan from "Regression test requirements for existing software which is modified" to "Regression test requirements for previously qualified software to be modified".

— Added a statement to Section 1.3 that project specific quality requirements shall be documented either in a Project Quality Plan by the EPM.

— In Section 6.3.2, changed "Others as deemed appropriate by the EPM" to "V&V team leader" for SCR approval/rejection routing. Also, changed the associated line in EXHIBIT 6-1 SOFTWARE CHANGE REQUEST FORM.

— Changed the responsibility for performing the Physical Review from the V&V team to the design team in Section 6.3.4.

— Clarified Section 4.6.2.10 by adding the following sentence: "Suggestions for improvement and/or best practices that are identified during the Post Mortem Review should be documented via EXHIBIT 9-2 CORRECTIVE ACTIONS PROCESS.

— Clarified Section 5.4.5.4 to state that "V&V database reviews are documented by completing the appropriate sections of EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST." Also, added data accuracy and data completeness line items to the checklist.

— Clarified Section 3.3.6 by stating that, "The Preliminary Hazards Analysis Report can be either completed by the V&V team or completed by the design team and reviewed by the V&V team."

— Changed EXHIBIT 4-1 ASSIGNMENT OF COMMON Q SOFTWARE TO CLASSES due to changes in configuration/terminology as follows:

- In the Plant/Reactor Protection System row, changed "Safety Critical Kernel (LCL, Bistable, CCC)" to "Safety Critical Kernel (LCL and Bistable)". This is a carry-over from an older design. There is no application software for Cross Channel Communications.

- Deleted "CIP Software" This was a separate Communication Interface Processor. This function is handled by the ITP.

- "Safety Critical Kernel (Pump and Valve)" was replaced with "Safety Critical Kernel (ILP)". ILP was defined as the "Integrated Logic Processor" in the table of acronyms.

— Clarified Section 6.2.2 by removing items that are not part of SCM responsibilities and adding some SCM responsibilities that were missing from the previous revision.

— Added a source file name to the program header block example in Section 6.3.1.

— Expanded EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST to include more detail relative to the review of software module and unit testing.

- Implemented changes based on NRC review and audit experience.

  — Provided a clear definition of responsible groups and organizational structure.

    - Defined V&V Group responsibilities as software design verification, software validation testing and software configuration management. See Sections 4.3.1, 6.2.2, 6.2.2.3. Added the following sentence to Section 4.3.2, "A Software Librarian maintains a configuration status accounting of software to effectively manage the software configuration."

    - Revised the process flow diagram (EXHIBIT 4-2 COMMON Q SOFTWARE DEVELOPMENT PROCESS) to clarify the interfaces and responsibilities of the Automation Engineering groups affected by the SPM. Also, removed the word "typical" from the title of the exhibit.

    - Revised EXHIBIT 1-1 RELATIONSHIP OF SPM TO IEEE STANDARDS (previously titled SOFTWARE PROGRAM DOCUMENTATION HIERARCHY) to clarify the software design and quality assurance activities required by the SPM and to illustrate the relationships among the standards referenced in the SPM for additional clarification. Added a paragraph to Section 1.1 to explain the Exhibit.

  — Expanded Section 5.4.5.3, Requirements Traceability Analysis, to place more emphasis on the importance of the Requirements Traceability Matrix (RTM). Other RTM related changes to SECTION 5 include:

    - Define the minimum acceptable requirements for the RTM.

    - The RTM shall be maintained as an historical document, i.e., requirements that have been deleted should be indicated by line out or other means to preserve the history.

    - The RTM can be prepared either manually or by using an automated tool.

    - The design team is responsible for performing the Requirements Traceability Analysis including preparation and updating the RTM. V&V team verifies the adequacy and accuracy.

- Author's signature on document signifies that the RTM has also been updated. (SECTION 8)

- Added a line item to the V&V checklists to ensure that the RTM is updated during each phase of the software life cycle and to remind the reviewer to verify the adequacy and accuracy of the RTM.

— Added a section titled "Documentation Requirements" to distinguish between requirements for documents and requirements for information to be included in a document. This section includes two tables. The first table includes a list of all documents required and the group or person responsible for preparing the document. The second table provides a list containing the section number where the requirement for information is identified, a description of the requirement, and the output document that satisfies the requirement.

— Add clarification in Sections 8.2.2 and 8.3 that the SRS and SDD shall be prepared using the IEEE standards for guidelines.

- Several miscellaneous changes were made to correct typographical errors and to clarify some of the process descriptions. For example:

— Added the following sentence to Archival Requirements, Section 4.11.2: "The requirements in this section are the responsibility of the software librarian and should be performed in accordance with Reference 4."

— Changed "Validated SCR" to "Approved SCR" in Section 7.3.1.

— Changed "Any safety and security issues shall be identified in these documents" to "Any safety and security requirements shall be included in these documents" in Section 7.3.2.2.

— Provided clarification in Section 8.2.2 that the SRS includes (rather than identify) software safety requirements that address system safety requirements.

— Changed "In-process audits shall be performed by the Quality organization to verify the consistency of the design and . . ." to "In-process audits shall be performed by the Quality organization to verify the consistency of the design process and . . ." in Section 6.3.4.

— Added the following sentence to Section 1.4.1 to clarify the software life cycle: "These phases may overlap or be performed iteratively."

— Clarified Section 4.12.2 by removing some redundant information.

— Added a paragraph to explain the relationship between Exception Reports and SCRs in the introduction to SECTION 7 SOFTWARE OPERATION AND MAINTENANCE PLAN.

— Removed sentence is Section 9.2 that specifically states that all exception report forms shall be placed in lifetime records. This specific reference to lifetime records is not necessary because all safety system software and documentation are designated as quality records and are controlled in accordance with Reference 4.

— Rewrote the V&V Organization description in Section 5.4.1 to provide additional clarification.

— Expanded the first sentence of SECTION 2, to describe an overview of the Westinghouse organizational structure.

— Combined Sections 3.1.1 and 3.1.2 into one section titled Purpose. The new section is 3.1. This is now more consistent with IEEE Std 1228-1994, "IEEE Standard For Software Safety Plans".

— Changed "lead verifier" to "V&V team leader" in Section 5.4.3.2.1 to be consistent with other sections of the SPM.

Revision 1A **December 2004**

Description:
This document is being prepared to create the NRC accepted version. The previously released document, WCAP-16096-NP-A, Revision 1, has been modified as follows:

- In accordance with established NRC procedures, this accepted version of the Common Q Software Program Manual includes the safety evaluation and associated cover letter that were issued by the NRC.
- Correction of typographical errors.
- Minor document format changes were made.

TABLE OF CONTENTS

Section	Title	Page
	LIST OF CONTRIBUTORS/REVIEWERS	i
	NRC SAFETY EVALUATION AND COVER LETTER – SEPTEMBER 28, 2004	iii
	REVISION HISTORY	v
	DETAILED RECORD OF CHANGES	vii
	TABLE OF CONTENTS	xix
	LIST OF TABLES	xxvii
	ACRONYMS AND TRADEMARKS	xxix
	GLOSSARY OF TERMS	xxxi
	REFERENCES	xxxiii
	DOCUMENTATION REQUIREMENTS	xxxv
SECTION 1	INTRODUCTION	1-1
1.1	PURPOSE	1-1
1.2	SCOPE	1-2
1.2.1	Software Classification and Categorization	1-2
1.2.2	Software Exclusions	1-3
1.3	OVERVIEW	1-3
1.4	GENERAL REQUIREMENTS	1-5
1.4.1	Software Life Cycle	1-5
1.4.2	Indoctrination and Training	1-5
SECTION 2	ORGANIZATION	2-1
SECTION 3	SOFTWARE SAFETY PLAN	3-1
3.1	PURPOSE	3-1
3.2	DEFINITIONS, ACRONYMS, ABBREVIATIONS AND REFERENCES	3-1
3.3	SOFTWARE SAFETY MANAGEMENT	3-1
3.3.1	Organization And Responsibilities	3-1
3.3.2	Resources	3-2
3.3.3	Staff Qualifications And Training	3-3
3.3.4	Software Life Cycle	3-5
3.3.5	Documentation Requirements	3-5
3.3.5.1	Software Project Management	3-5
3.3.5.2	Software Configuration Management	3-5
3.3.5.3	Software Quality Assurance	3-5

3.3.5.4 Software Safety Requirements 3-5
3.3.5.5 Software Design Description 3-5
3.3.5.6 Software Development Methodology, Standards, Practices, Metrics and Conventions 3-6
3.3.5.7 Test Documentation 3-6
3.3.5.8 Software Verification and Validation 3-6
3.3.5.9 Reporting Safety Verification and Validation 3-7
3.3.5.10 Software User Documentation 3-7
3.3.5.11 Results of Software Safety Requirements Analysis 3-7
3.3.5.12 Results of Software Safety Design Analysis 3-7
3.3.5.13 Results of Software Safety Code Analysis 3-7
3.3.5.14 Results of Software Safety Test Analysis 3-7
3.3.5.15 Results of Software Safety Change Analysis 3-7
3.3.6 Software Safety Program Records 3-8
3.3.7 Software Configuration Management Activities 3-8
3.3.8 Software Quality Assurance Activities 3-9
3.3.9 Software Verification and Validation Activities 3-9
3.3.10 Tool Support and Approval 3-9
3.3.11 Previously Developed Or Purchased Software 3-9
3.3.12 Subcontract Management 3-9
3.3.13 Process Certification 3-9
3.4 SOFTWARE SAFETY ANALYSES 3-10
3.4.1 Software Safety Analyses Preparation 3-10
3.4.2 Software Safety Requirements Analysis 3-11
3.4.3 Software Safety Design Analysis 3-11
3.4.4 Software Safety Code Analysis 3-11
3.4.5 Software Safety Test Analysis 3-11
3.4.6 Software Safety Change Analysis 3-11
3.5 POST DEVELOPMENT 3-11
3.5.1 Training 3-12
3.5.2 Deployment 3-12
3.5.2.1 Installation 3-12
3.5.2.2 Startup and Transition 3-12
3.5.2.3 Operations Support 3-13
3.5.3 Monitoring 3-13
3.5.4 Maintenance 3-13
3.5.5 Retirement And Notification 3-13
3.6 PLAN APPROVAL 3-13

SECTION 4 SOFTWARE QUALITY ASSURANCE PLAN 4-1

4.1 PURPOSE 4-1
4.1.1 Introduction 4-1
4.1.2 Scope 4-1
4.1.3 Software Development Process 4-3
4.2 REFERENCES 4-3
4.3 MANAGEMENT 4-3
4.3.1 Organization 4-4
4.3.2 Tasks and Responsibilities 4-5
4.3.2.1 Initiation Phase 4-5
4.3.2.2 Software Requirements Phase 4-5
4.3.2.3 Software Design Phase 4-6
4.3.2.4 Software Implementation Phase 4-7
4.3.2.5 Testing Phase 4-8
4.3.2.6 Site Installation and Checkout Phase 4-8
4.3.2.7 Operations and Maintenance Phase 4-9
4.4 DOCUMENTATION 4-9
4.4.1 Purpose 4-9
4.5 STANDARDS, PRACTICES, CONVENTIONS AND METRICS 4-10
4.5.1 Purpose 4-10
4.5.2 Content 4-10
4.5.2.1 Coding Standards 4-10
4.5.2.2 Software Testing Standards 4-11
4.5.2.3 Documentation Standards 4-11
4.5.2.4 Metrics 4-11
4.6 REVIEWS 4-12
4.6.1 Purpose 4-12
4.6.2 Minimum Requirements 4-13
4.6.2.1 Software Requirements Review (SRR) 4-13
4.6.2.2 Software Design Review 4-14
4.6.2.3 Code Verification 4-15
4.6.2.4 Software Verification and Validation Plan Review 4-15
4.6.2.5 Functional Review 4-15
4.6.2.6 Physical Review 4-16
4.6.2.7 In-Process Audits 4-16
4.6.2.8 Managerial Reviews 4-16
4.6.2.9 Software Configuration Management Plan Review 4-17
4.6.2.10 Post Mortem Review 4-17
4.7 TEST 4-17
4.8 PROBLEM REPORTING AND CORRECTIVE ACTION 4-17
4.8.1 Purpose and Scope 4-17

4.9 TOOLS, TECHNIQUES AND METHODOLOGIES 4-18
4.10 CODE CONTROL 4-18
4.11 MEDIA CONTROL 4-19
4.11.1 Media Identification 4-19
4.11.2 Archival Requirements 4-19
4.12 SUPPLIER CONTROL 4-19
4.12.1 Existing Software 4-19
4.12.2 Sub-Contracted Software/Services 4-20
4.13 RECORDS COLLECTION, MAINTENANCE AND RETENTION 4-21
4.14 TRAINING 4-21
4.15 RISK MANAGEMENT 4-21

SECTION 5 SOFTWARE VERIFICATION AND VALIDATION PLAN 5-1
5.1 PURPOSE 5-1
5.1.1 Categorization of Software Items and Review Scope 5-1
5.1.2 V&V Program Implementation 5-1
5.1.3 Prominence of V&V Documentation 5-2
5.1.4 Overall Common Q and System Specific V&V Plans 5-2
5.2 REFERENCED DOCUMENTS 5-2
5.3 DEFINITIONS 5-3
5.4 VERIFICATION AND VALIDATION OVERVIEW 5-3
5.4.1 Organization 5-3
5.4.2 Master Schedule 5-3
5.4.3 Resources summary 5-4
5.4.3.1 Design Team 5-4
5.4.3.2 Verification and Validation Team 5-5
5.4.4 Responsibilities 5-6
5.4.4.1 Verification and Validation Team Responsibilities 5-6
5.4.5 Tools, Techniques and Methodologies 5-6
5.4.5.1 Automated Tools 5-6
5.4.5.2 V&V Core Activities 5-6
5.4.5.3 Requirements Traceability Analysis 5-8
5.4.5.4 Database Review/Testing 5-9
5.5 LIFE-CYCLE VERIFICATION AND VALIDATION 5-10
5.5.1 Management of V&V 5-10
5.5.2 Concept (Initiation) Phase V&V 5-11
5.5.2.1 V&V Inputs 5-11
5.5.2.2 V&V Tasks 5-11
5.5.2.3 V&V Outputs 5-11
5.5.3 Requirements Phase V&V 5-12

5.5.3.1 V&V Inputs 5-12
5.5.3.2 V&V Tasks 5-13
5.5.3.3 V&V Outputs 5-16
5.5.4 Design Phase V&V 5-16
5.5.4.1 V&V Inputs 5-16
5.5.4.2 V&V Tasks 5-17
5.5.4.3 V&V Outputs 5-17
5.5.5 Implementation Phase V&V 5-18
5.5.5.1 V&V Inputs 5-18
5.5.5.2 V&V Tasks 5-18
5.5.5.3 V&V Outputs 5-19
5.5.6 Test Phase V&V 5-20
5.5.6.1 V&V Inputs 5-21
5.5.6.2 V&V Tasks 5-22
5.5.6.3 V&V Outputs 5-23
5.5.7 Installation and Checkout Phase V&V 5-24
5.5.7.1 V&V Inputs 5-24
5.5.7.2 V&V Tasks 5-24
5.5.7.3 V&V Outputs 5-25
5.5.8 Operation and Maintenance Phase V&V 5-25
5.6 SOFTWARE VERIFICATION AND VALIDATION REPORTING 5-26
5.6.1 Required Reports 5-26
5.6.2 Optional Reports 5-27
5.7 VERIFICATION AND VALIDATION ADMINISTRATIVE PROCEDURES 5-27
5.7.1 Anomaly Reporting And Resolution 5-27
5.7.2 Task Iteration Policy 5-27
5.7.3 Deviation Policy 5-27
5.7.4 Control Procedures 5-27
5.7.5 Standards, Practices, And Conventions 5-27
5.8 V&V TEST DOCUMENTATION REQUIREMENTS 5-28
5.8.1 Test Plan 5-28
5.8.2 Test Procedure 5-28
5.8.2.1 Test-Design Specification 5-28
5.8.2.2 Test-Case Specification 5-28
5.8.2.3 Test-Procedure Specification 5-28
5.8.3 Test Report 5-28
5.9 SOFTWARE INTEGRITY LEVEL SCHEME 5-29

SECTION 6 SOFTWARE CONFIGURATION MANAGEMENT PLAN 6-1
6.1 INTRODUCTION 6-1

6.1.1 Purpose 6-1
6.1.2 Scope 6-2
6.1.3 Definitions 6-3
6.1.4 References 6-3
6.2 MANAGEMENT 6-3
6.2.1 Organization 6-3
6.2.2 SCM Responsibilities 6-3
6.2.2.1 Requirement Phase 6-3
6.2.2.2 Design Phase 6-4
6.2.2.3 Implementation Phase 6-4
6.2.2.4 Test Phase 6-4
6.2.2.5 Installation and Checkout Phase 6-5
6.2.2.6 Operations and Maintenance Phase 6-5
6.2.2.7 Retirement Phase 6-5
6.2.2.8 Configuration Identification Management 6-6
6.2.2.9 Configuration Control Management 6-6
6.2.2.10 Configuration Status Accounting Management 6-6
6.2.2.11 Configuration Reviews and Audits 6-6
6.2.3 Applicable Policies, Directives and Procedures 6-6
6.3 SOFTWARE CONFIGURATION MANAGEMENT ACTIVITIES 6-6
6.3.1 Configuration Identification 6-6
6.3.2 Configuration Change Control 6-9
6.3.3 Configuration Status Accounting 6-11
6.3.4 Configuration Audits And Reviews 6-11
6.3.5 Interface Control 6-12
6.3.6 Subcontractor/Vendor Control 6-13
6.3.6.1 Subcontractor Software 6-13
6.3.6.2 Vendor Software 6-13
6.4 SCM SCHEDULES 6-14
6.5 SCM RESOURCES 6-14
6.6 SCM PLAN MAINTENANCE 6-14

SECTION 7 SOFTWARE OPERATION AND MAINTENANCE PLAN 7-1
7.1 INTRODUCTION 7-1
7.2 PROBLEM/MODIFICATION IDENTIFICATION, CLASSIFICATION AND PRIORITIZATION 7-1
7.2.1 Input 7-1
7.2.2 Process 7-1
7.2.3 Control 7-2
7.2.4 Output 7-2

7.3 ANALYSIS 7-2
7.3.1 Analysis Input 7-2
7.3.2 Analysis Process 7-2
7.3.2.1 Feasibility Analysis 7-2
7.3.2.2 Detailed Analysis 7-3
7.3.3 Analysis Control 7-3
7.3.4 Analysis Output 7-4
7.4 DESIGN 7-4
7.4.1 Design Input 7-4
7.4.2 Design Process 7-4
7.4.3 Design Control 7-4
7.4.4 Design Output 7-5
7.5 IMPLEMENTATION 7-5
7.5.1 Implementation Input 7-5
7.5.2 Implementation Process 7-5
7.5.2.1 Coding and Module Testing 7-5
7.5.2.2 Integration 7-5
7.5.2.3 Documentation 7-5
7.5.2.4 Risk Analysis And Test-Readiness Review 7-6
7.5.3 Implementation Control 7-6
7.5.4 Implementation Output 7-6
7.6 TEST 7-6
7.6.1 Test Input 7-6
7.6.2 Test Process 7-6
7.6.3 Test Control 7-7
7.6.4 Test Output 7-7
7.7 DELIVERY 7-7
7.7.1 Input 7-7
7.7.2 Process 7-7
7.7.3 Control 7-7
7.7.4 Output 7-7

SECTION 8 DOCUMENTATION 8-1
8.1 GENERAL REQUIREMENTS 8-1
8.2 SYSTEM REQUIREMENTS DOCUMENTATION 8-1
8.2.1 System Requirements Specification (SysRS) 8-1
8.2.2 Software Requirements Specification (SRS) 8-2
8.3 SOFTWARE DESIGN DESCRIPTION (SDD) 8-2
8.4 SOURCE CODE DOCUMENTATION 8-3
8.5 SOFTWARE VERIFICATION AND VALIDATION DOCUMENTATION 8-3

8.5.1 Software Verification and Validation Plan 8-3
8.5.2 Software Verification and Validation Report 8-4
8.6 USER DOCUMENTATION 8-4
8.7 SOFTWARE CONFIGURATION MANAGEMENT DOCUMENTATION 8-4
8.8 TEST DOCUMENTATION 8-4
8.8.1 Test Plans 8-5
8.8.2 Test Procedures 8-5
8.9 COMPUTER CODE CERTIFICATE 8-5

SECTION 9 PROBLEM REPORTING AND CORRECTIVE ACTION 9-1
9.1 INTRODUCTION 9-1
9.2 ERROR REPORTING BEFORE SOFTWARE RELEASE 9-1
9.3 ERROR REPORTING AFTER SOFTWARE RELEASE 9-2
9.4 CORRECTIVE ACTION 9-2

SECTION 10 EXHIBITS 10-1
EXHIBIT 1-1 RELATIONSHIP OF SPM TO IEEE STANDARDS 10-3
EXHIBIT 2-1 DESIGN/V&V TEAM ORGANIZATION 10-4
EXHIBIT 4-1 ASSIGNMENT OF COMMON Q SOFTWARE TO CLASSES 10-5
EXHIBIT 4-2 COMMON Q SOFTWARE DEVELOPMENT PROCESS 10-7
EXHIBIT 4-3 TASKS REQUIRED FOR SOFTWARE CATEGORIES 10-8
EXHIBIT 4-4 COMMON Q COMMENT RECORD 10-11
EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES 10-12
EXHIBIT 5-2 CHECKLIST NO. 1, SOFTWARE VERIFICATION AND VALIDATION REQUIREMENTS PHASE CHECKLIST 10-14
EXHIBIT 5-3 CHECKLIST NO. 2, SOFTWARE VERIFICATION AND VALIDATION DESIGN PHASE CHECKLIST 10-18
EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST 10-20
EXHIBIT 5-5 CHECKLIST NO. 4, SOFTWARE VERIFICATION AND VALIDATION TEST PHASE CHECKLIST 10-23
EXHIBIT 5-6 CHECKLIST NO. 5, SOFTWARE VERIFICATION AND VALIDATION INSTALLATION AND CHECKOUT PHASE CHECKLIST 10-26
EXHIBIT 6-1 SOFTWARE CHANGE REQUEST FORM 10-27
EXHIBIT 8-1 COMPUTER CODE CERTIFICATE 10-28
EXHIBIT 9-1 EXCEPTION REPORT 10-29
EXHIBIT 9-2 CORRECTIVE ACTIONS PROCESS 10-30

TABLE OF CONTENTS (Cont'd)

LIST OF TABLES

Table	Title	Page
Table I.	Document Requirements	xxxv
Table II.	Information Requirements	xxxvi
Table 3.3.3-1.	Software Safety Task Assignments	3-3
Table 5.9-1.	Software Classification Mapping	5-29
Table 6.3.1-1.	Media Identification Example	6-9
Table 9.2-1.	Error Reporting Methods	9-1

This Page Intentionally Left Blank

ACRONYMS AND TRADEMARKS

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

Acronyms	Definition
ABB	Asea Brown Boveri
ANSI	American National Standards Institute
BTP	Branch Technical Position
CAPS	Westinghouse Corrective Actions Process
CCC	Cross Channel Communications
CDR	Critical Design Review
CET	Core Exit Thermocouple
CFR	Code of Federal Regulations
CGDP	Commercial Grade Dedication Program
CM	Configuration Management
CMS	Configuration Management System
COTS	Commercial Off-The-Shelf
CPC	Core Protection Calculator
CPCS	Core Protection Calculator System
CPU	Central Processing Unit
DESFAS-AC	Digital Engineered Safety Features Actuation System Auxiliary Cabinet
DPPS	Digital Plant Protection System
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
IEEE	Institute of Electrical and Electronics Engineers
I&C	Instrumentation and Control
I/O	Input and Output
ILP	Integrated Logic Processor
ITP	Interface and Test Processor
LAN	Local Area Network
LCL	Local Coincidence Logic
LN	Lotus Notes (IBM database management system)

NA	Nuclear Automation
NPP	Nuclear Power Plant
NQA	Nuclear Quality Assurance
NuCAR	Nuclear Classification Assignment Record
PAMS	Post Accident Monitoring System
PDR	Preliminary Design Review
PHA	Preliminary Hazards Analysis
PPS	Plant Protection System
PQP	Project Quality Plan
QMS	Quality Management System
RPS	Reactor Protection System
RRAS	Repair, Replacement and Automation Services
RTA	Requirements Traceability Analysis
RTM	Requirements Traceability Matrix
RVL	Reactor Vessel Level
SAT	Site Acceptance Test
SCA	Software Change Authorization
SCM	Software Configuration Management
SCMP	Software Configuration Management Plan
SCR	Software Change Request
SDD	Software Design Description
SM	Subcooled Margin
SOMP	Software Operation and 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
V&V	Verification and Validation
VT	Verification and Validation Team

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
Advant	Advant is a registered trademark of ABB Process Automation Corp. The Common Q platform includes the Advant Controller 160 (AC160). The AC160 is part of the ABB Advant Power system. It is used in applications that require high availability and redundancy.
Automation Engineering	Automation Engineering is the cognizant engineering organization within Westinghouse Electric Company that is responsible for the design and implementation of Common Q based systems.
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 ensure 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, 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. 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.
Module	A module is the smallest software entity that is subjected to testing. It is a custom PC element or a type circuit in Advant space or a subroutine in language programming space.

Platform Lead	The Common Q Platform Lead is responsible to ensure that the platform development meets the continuing needs of the product family.
Quality	"Quality" is the generic title of the independent quality assurance department that is responsible for coordinating and reviewing quality assurance procedures and directives. The Quality organization is part of the Repair, Replacement and Automation Services business unit; but, it is separate from the Automation Engineering organization. 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.
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."
Software Item	A software item is defined as the 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	A unit consists of several modules that are integrated into a separately testable element, logically consistent with design specifications. It is a type circuit or control module in Advant 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 Quality Management System

2. ASME NQA-1-1994, 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. Nuclear Automation Edition of the Westinghouse Policies and Procedures Manual

5. IEEE Std 610.12-1990, "IEEE Standard Glossary of Software Engineering Terminology."

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

7. IEEE Std 1016-1998, "IEEE Recommended Practice for Software Design Descriptions"

8. IEEE Std 1012-1998, "IEEE Standard for Software Verification and Validation"

9. IEEE Std 1063-1987, "IEEE Standard for Software User Documentation"

10. IEEE Std 828-1998, "IEEE Standard for Software Configuration Management Plans"

11. IEEE Std 7-4.3.2-1993, "IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations"

12. IEEE Std 1008-1987, "IEEE Standard for Software Unit Testing"

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 1219-1998, "IEEE Standard For Software Maintenance"

16. IEEE Std 1028-1997, "IEEE Standard For Software Reviews"

REFERENCES (Cont'd)

17. Reg. Guide 1.152, Rev. 1 (1996), "Criteria for Digital Computers in Safety Systems of Nuclear Power Plants"

18. Reg. Guide 1.168, Rev. 1 (Feb. 2004), "Verification, Validation, Reviews, and Audits for Digital Computer Software used in Safety Systems of Nuclear Power Plants"

19. Reg. Guide 1.169, Rev. 0 (Sept. 1997), "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. 0 (Sept. 1997), "Software Unit Testing for Digital Computer Software used in Safety Systems of Nuclear Power Plants"

22. Reg. Guide 1.172, Rev. 0 (Sept. 1997), "Software Requirements Specifications for Digital Computer Software used in Safety Systems of Nuclear Power Plants"

23. Reg. Guide 1.173, Rev. 0 (Sept. 1997), "Developing Software Life Cycle Processes For Digital Computer Software Used In Safety Systems Of Nuclear Power Plants"

24. IEEE Std 1074-1997, "IEEE Standard for Developing Software Life Cycle Processes"

25. (Reference Deleted)

26. IEEE Std 1228-1994, "IEEE Standard For Software Safety Plans"

27. NUREG-0800, "USNRC Standard Review Plan for the Review of Safety Analysis Reports for Nuclear Power Plants", (SRP) Chapter 7, Revision 4, dated June 1997

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

29. (Reference Deleted)

30. (Reference Deleted)

31. WCAP-7211, "Proprietary Information and Intellectual Property Management"

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	Audit Report (In-Process Audit)	Quality
2	Certificate Of Conformance (System)	Quality
3	Code Review Report	Independent Reviewer from either the Design Team or V&V Team
4	Coding Standards and Guidelines	Design Team
5	Commercial Grade Dedication Report	Design Team or V&V Team
6	Exception Report (Database)	V&V Team or Design Team
7	Exception Report Log (Database)	V&V Team or Design Team
8	Failure Modes and Effects Analysis	Design Team
9	Nuclear Classification Assignment Record (NuCAR)	Design Team
10	Preliminary Hazards Analysis Report	V&V Team or Design Team
11	Project Quality Plan (Identifies supplemental quality requirements for a project.)	Design Team
12	Project Schedule	Design Team
13	Purchase Order	Customer
14	Regression Analysis	Design Team or V&V Team
15	Requirements Traceability Matrix (Database)	Design Team
16	Resource Plan	EPM
17	Software Change Request (Database)	Design Team or V&V Team
18	Software Change Request Log (Database)	Design Team or V&V Team
19	Software Design Description	Design Team

Table I. Document Requirements

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

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
3.1.1	Defining Acceptable Risks	Project Quality Plan
3.3.2	A detailed schedule	SAP
3.3.2	Resource Plan	SAP

Table II. Information Requirements

SPM Section Number	Description of Requirement	Output Document
3.3.5.10	Software User Documentation	Technical Manual
3.3.5.11	Results of Software Safety Requirements Analysis	V&V Report
3.3.5.12	Results of Software Safety Design Analysis	V&V Report
3.3.5.13	Results of Software Safety Code Analysis	V&V Report
3.3.5.14	Results of Software Safety Test Analysis	V&V Report
3.3.5.15	Results of Software Safety Change Analysis	V&V Report
3.3.6	Software Hazards	Preliminary Hazards Analysis Report
3.3.6	Results of V&V Analyses	V&V Report
3.3.6	Information on suspected or confirmed safety problems	V&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	V&V Report
3.3.6	Tracking system to ensure hazards and their status 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 Quality Plan
3.4.1	Preliminary Hazards Analysis	Preliminary Hazards Analysis Report
3.4.2	Software Safety Requirements Analysis	V&V Report
3.4.3	Software Safety Design Analysis	V&V Report
3.4.4	Software Safety Code Analysis	V&V Report

Table II. Information Requirements

SPM Section Number	Description of Requirement	Output Document
3.4.5	Software Safety Test Analysis	V&V Report
3.4.6	Software Safety Change Analysis	V&V Report
3.5	Training in SPM Section 9	Training Record
3.5.1	Review of Training Materials	V&V Report
3.5.1	Personnel Training	Training Record
3.5.2.1	Review of Installation documentation	V&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	Nuclear Classification Assignment Record (NuCAR)
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	LN Database
4.3.2.6	Exception Report	LN Database
4.5.1	Work Instructions	Any document required to supplement the SPM (e.g. Coding Standards and Guidelines Document)
4.5.2.1	Coding Standards	Coding Standards and Guidelines Document
4.5.2.2	Metric Reporting	Test Reports
4.6.2.1	Software Requirements Review	V&V Report
4.6.2.2.1	Preliminary Design Review	V&V Report

Table II. Information Requirements

SPM Section Number	Description of Requirement	Output Document
4.6.2.2.2	Critical Design Review	V&V Report
4.6.2.3	Code Certification	Code Review Reports
4.6.2.4	SVVP Review	SPM
4.6.2.5	Functional Review	V&V Report
4.6.2.6	Physical Review	Certificate of Conformance
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	V&V Report
4.6.2.10	Post Mortem Review	CAPS (LN Database)
5.1.4	Project Specific V&V Plan Activities	Project Quality Plan
5.4.5.2	V&V Checklists	V&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 #4)	Implementation Phase Checklist in V&V Report
5.5.1	Baseline Change Assessment	Regression Analysis
5.5.4.3	V&V Outputs, #2	V&V Report
5.5.6	Software Safety Test Analysis	Test Phase Checklist in V&V Report
5.5.6.3	Code Certificate	V&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

Table II. Information Requirements

SPM Section Number	Description of Requirement	Output Document
		Requirements
5.5.8	Regression Analysis	V&V Report or separately prepared document
5.6.1	Discrepancy Reports	Comment Record Database; Status defined in V&V Report
6.2.2.1	Define software items which are to be controlled via SCM	Project Quality Plan
6.3.2	Master list of software under configuration control for a project	Software Release Record
6.3.2	Software Change Request	LN Database
6.3.2	Software Change Request Log	LN Database
6.3.3	Configuration Status Accounting	Software Release Record
7.3.2.1	Feasibility Analysis	Project Quality Plan
7.3.2.2	Detailed Analysis	SysRS, SRS, Test Plan, PQP
7.5.2.4	Risk Analysis	Project Quality Plan
9.2	Justification for not performing complete system testing	Software Release Record
9.2	Exception Reports	LN 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 Policies and Procedures Manual (Reference 4). This manual is compliant with (ASME) NQA-1-1994, Subpart 2.7, and ISO 9000-3 Tick-IT.

Additions, deletions and alternatives to the SPM requirements, in order to be compliant with IEC 880, are defined in a separate document.

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, problem 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 Operations and Maintenance Plan, which describes software practices after delivery to a customer.

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 Std 603 and IEEE Std 7-4.3.2 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 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 systems and modules are identified as belonging to one of the following classes:

- **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 on-line plant system.

The requirements for the classification of functions, systems and equipment are provided in Reference 4. There is an approximate equivalence between Protection, Important-to-Safety, Important-to-Availability,

and General Purpose, and Categories A, B, C and O in NA 4.4.1, respectively. The classifications shown in EXHIBIT 4-1 ASSIGNMENT OF COMMON Q SOFTWARE TO CLASSES, at the system-function level illustrate a consistent agreement.

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 Nuclear Classification Assignment Record (NuCAR) 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

Every Common Q software item is assigned to one of the above categories. 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 Excel, Word, and AutoCAD.

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). 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 in a Project Quality Plan. 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 will ensure the verification of an accurate translation from one step in the software development process to the next step and the validation that the software product does fulfill 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 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.

Recommendations are made for a Software Operation and Maintenance Plan to be administered by others.

The recommendations for the Software Operations and Maintenance Plan describe the activities necessary to maintain the Common Q software, to remove latent errors, to respond to new or revised requirements and to adapt the software to changes in operating environments.

The documentation section 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 of the Software Program Manual describes procedures necessary to ensure 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.

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. The SPM meets the requirements of Reference 11 as augmented by Reference 17.

1.4.1 Software Life Cycle

This Software Program Manual is based on a Software Life Cycle model consistent with References 2 and 5 which includes the following phases:

- 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.

This Page Intentionally Left Blank

(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 to ensure contractual requirements are identified and met, a focal point for assuring 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 Repair, Replacement and Automation Services (RRAS) organization. Within this organization, software activities are organized into the following two teams:

- The design team develops the system requirements, software design, and code for the Common Q systems. The design team may develop common software that is used in systems developed by other groups.
- The V&V team performs software design verification, software validation testing and software configuration management activities on the Common Q systems. The design team may perform the validation testing activities. In this case, the V&V team shall oversee the conduct of these activities by reviewing documentation and witnessing testing.

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

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 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 any software defects do not present severe consequences to public health and safety.

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 Quality Plan for a given system implementation.

This plan is prepared in accordance with Reference 27, Branch Technical Position (BTP) HICB-14, "Guidance on Software Reviews for Digital Computer-Based Instrumentation and Control Systems", and Reference 26. 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 xxix for a list of acronyms and abbreviations. Refer to page xxxi for definitions. Refer to page xxxiii for a list of references.

3.3 SOFTWARE SAFETY MANAGEMENT

Consistent with Reference 26 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 two parts:

1. The Quality organization, an independent quality assurance department, coordinates and reviews quality assurance procedures and directives. The Quality organization is part of the Repair, Replacement and Automation Services business unit; but, it is separate from the Automation Engineering organization. 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 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 Automation Engineering organization performs the safety activities for a given Common Q system implementation project. Refer to Section 5.4.3.2 for a description of the V&V team.

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

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

2. Coordinate software safety task planning and implementation with the design team per SECTION 5.

3. Ensure 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

Automation Engineering 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 V&V team leader shall determine the resources required to implement a Common Q system. ELMs shall assign the appropriate resources to the EPM and V&V team leader. The following resources are considered for both the design and V&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.

A detailed schedule with linked tasks and work breakdown structure shall include software safety activities and be maintained to monitor progress throughout the project. Project schedules and resource allocations are established and maintained in SAP.

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 performing the software design.

The following table identifies the personnel that will perform the tasks identified in Reference 26, Section 4.3.3:

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	V&V team
Test safety-critical features	V&V team[1]
Audit software safety plan implementation	Quality organization
Perform process certification	Quality organization

One of the most important factors in developing reliable software is the development and use of a qualified staff. Protection class software needs to be prepared by carefully selected personnel who can produce a reliable product. Although staff selection is one means of establishing a capable staff, training

[1] The V&V team can delegate the responsibility of testing to the design team. In this case, the V&V team shall review the test plans, procedures and execution for compliance.

is also a key factor in establishing and maintaining 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 needs need to be periodically reassessed.

The V&V team shall be trained in the tools, techniques and methodologies described in Section 5.4.5.

The ELM assures 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. These changes, which are sometimes subtle, may result in previously qualified personnel being unqualified for the changed work. Therefore, 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 have 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 V&V is defined in Section 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 8, 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 ensure that all are done according to prescribed procedures and that adequate resources are allocated 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 8.2) 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 8.3) 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 all tests that will be conducted for the Common Q project. They shall contain the requirements for all acceptance test procedures and defines each required test to be conducted. They also define the methodology for the disposition of test exceptions (errors). This document is verified against the outputs generated from the requirements phase of V&V for completeness. All prerequisites for testing shall also be identified. Section 4.3.2.2 describes the requirements for a test plan.

3.3.5.7.2 Test Procedures

The test procedures are the actual tests conducted on the Common Q software. They include test setup, precautions and limitations, 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 V&V. Refer to Section 5.5.6 for a description of test procedure contents.

3.3.5.7.3 Test Reports

The test reports document the execution of the acceptance test procedures. In addition to attaching the signed and checked off test procedure, 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 Reference 14, Section 10.

3.3.5.8 Software Verification and Validation

The Software V&V documentation is described in SECTION 5.

3.3.5.9 Reporting Safety Verification and Validation

V&V reporting is described in SECTION 5.

3.3.5.10 Software User Documentation

User documentation is described in Section 8.6.

3.3.5.11 Results of Software Safety Requirements Analysis

The results of the Software Safety Requirements Analysis as described in Section 3.4.2 below shall be documented in the Requirements Phase section of the V&V Report (Section 8.5).

3.3.5.12 Results of Software Safety Design Analysis

The results of the Software Safety Design Analysis as described in Section 3.4.3 below shall be documented in the Design Phase section of the V&V Report (Section 8.5).

3.3.5.13 Results of Software Safety Code Analysis

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

3.3.5.14 Results of Software Safety Test Analysis

The results of the Software Safety Test Analysis as defined in Section 3.4.5 below shall be found in the V&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 Section 3.4.6 below shall be found in the V&V Report. For each software life cycle that is revisited by the design team, the V&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 V&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 8.

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 Preliminary Hazards Analysis Report as described in Section 3.4.1. Each software producible hazard is evaluated during each phase of development of the safety critical software. The Preliminary Hazards Analysis Report can be either completed by the V&V team or completed by the design team and reviewed by the V&V team.

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 prerelease or installed system is recorded in the Final V&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 Automation Engineering 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 ensure 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 V&V team leader approves 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 inadvertent introduction of software hazards by project tools is mitigated by the proper use of techniques for software configuration management, software quality assurance and V&V as described in this SPM.

3.3.11 Previously Developed Or Purchased Software

Section 4.1.2 describes the requirements for using existing software, including purchased software, as safety critical software.

3.3.12 Subcontract Management

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

3.3.13 Process Certification

A Certificate of Conformance 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.

A Preliminary Hazards Analysis (PHA) 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 Preliminary 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 (Section 8.2.1) provides the high-level system design as required in Section 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 (Section 8.2.2).

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 Section 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 Section 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 Section 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 Safety Test Analysis

Refer to Sections 5.5.5 and 5.5.6 for a description of the software safety test analyses performed for module/unit testing and system testing respectively. These activities provide reasonable assurance that each system and software safety requirement is tested.

3.4.6 Software Safety Change Analysis

Refer to Section 5.5.8 and SECTION 7 for a description of the software safety change analyses performed. 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 9 involving problem 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 V&V team per Section 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 V&V team shall review this documentation according to the procedure in Section 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 8. This documentation includes design documents, user manuals and instructions for maintenance expected by plant personnel.

3.5.3 Monitoring

The Software Operations and Maintenance Plan (SECTION 7) contains requirements for monitoring the use of delivered software and associated problem 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 Operational and Maintenance Plan in SECTION 7.

3.5.5 Retirement And Notification

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

3.6 PLAN APPROVAL

See SPM signature page for Software Safety Plan approval.

This Page Intentionally Left Blank

(Last Page of Section 3)

SECTION 4 SOFTWARE QUALITY ASSURANCE PLAN

4.1 PURPOSE

4.1.1 Introduction

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 Section 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 Nuclear Classification Assignment Record (NuCAR) 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 NuCAR in accordance with the Requirements of Reference 4.

NuCARs shall be prepared by the design team and reviewed by the V&V team.

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 2.

Within each software class described in Section 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 V&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 V&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
- Problem 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 (V&V) activities are governed by the Software V&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 xxxiii for a list of references.

4.3 MANAGEMENT

The management of all software for Common Q projects spans the software life cycle defined in Reference 2 and applies to all software classes described in Section 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 Repair, Replacement and Automation Service Organization (RRAS) described in SECTION 2. A design team, a V&V team, and a Quality organization within RRAS are responsible for the execution of all quality assurance tasks.

The design team and the V&V team are both within the Automation Engineering organization. The design team is responsible for the software design and implementation, and software quality assurance planning. The V&V team is responsible for software design verification, software validation testing and software configuration management.

The Quality organization is responsible for coordinating and reviewing quality assurance procedures and directives. The Quality organization is part of the RRAS business unit, but it is separate from the Automation Engineering organization. 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 Engineering Project Manager (EPM) shall ensure that all design team activities are 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 V&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 (Section 4.5.2.1), testing standards established in the test plan and documentation standards (SECTION 8) 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 Automation Engineering design and V&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 type of actions that are performed to ensure 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 author and the V&V team leader. If test procedures are generated by the design team and reviewed by the V&V team, the V&V phase summary report will be generated and signed by the V&V group leader to indicate the review and acceptance of these documents.

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

5. A software librarian maintains a configuration status accounting of software to effectively manage the software configuration.

4.3.2.1 Initiation 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 SOMP shall be specified in the PQP. The PQP author shall also define, or reference the applicable coding standards within the PQP.

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 Section 8.2.2 of this SPM.

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

A Common Q specific test plan shall start to be developed in accordance with Reference 14, Section 3, 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 and test cases, 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.

Test plan development continues into subsequent phases and is completed in the implementation phase.

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 8.3.

The V&V team as indicated in EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES shall verify each SDD. The verification review shall ensure 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 Section 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.

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 Section 4.5.2.1.

Existing software, which has been qualified as described in Section 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 Section 4.6.2.3.

The test plan, started during the software requirements phase, is completed in this phase.

Testing 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 a factory acceptance test. 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.

Module and unit testing shall be performed in accordance with the Test Plan and Reference 12. The responsibility for testing will be assigned to the design team and V&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.

4.3.2.5 Testing Phase

Factory acceptance testing shall be conducted during this phase in the development environment when all of the system components have been integrated by the design team per the 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 Reference 14, Sections 6 and 10 respectively, and verified by the groups identified in EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES. The groups identified in the exhibit shall conduct system tests. Also, test procedures and reports shall be developed in accordance with Reference 14, Sections 6 and 10 respectively, and consistent with the Test Plan.

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 8. Also, during this phase, all user documentation shall be verified by the V&V team.

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 V&V team shall be responsible for associated V&V requirements.

The preparation of the site test plan will be initiated during 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.

An Exception Report Log shall be maintained during the installation and checkout phase in accordance with the Site Acceptance Test (SAT) plan. For protection class software this log shall be verified by the V&V team after installation.

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 8.5.

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 8 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 ensure 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. Coding standards to be applied to a project shall be referenced in the Project Quality Plan. The V&V team shall review the applicable coding standards for each project for acceptability.

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 (with test cases) and test reports shall also be provided in the Test Plan and shall be consistent with Reference 14, Sections 6 and 10.

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 8.

4.5.2.4 Metrics

The following metrics should be maintained for each Common Q system:

1. The errors discovered during dry run of the FAT should be identified through the use of EXHIBIT 9-1 EXCEPTION REPORT forms so that all errors discovered can be resolved prior to FAT and that 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 pre-FAT to FAT to SAT. The exhibit represents the minimum information required. The exception reporting procedure shall be implemented via an automated process.

2. 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 Sections 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
  - — Preliminary 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 ensure 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. The intent of this section is to provide guidelines to conduct reviews and audits consistent with the requirements of Reference 16. In some cases, the procedural aspects of the review are contained in the SVVP.

4.6.2 Minimum Requirements

Reviews shall evaluate specific software elements (such as files, functions, modules, or complete systems) to ensure 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 V&V team shall conduct the Software Requirements Review (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 Preliminary Design Review

After the initial issuance of the SDDs, the V&V team shall conduct the Preliminary Design Review (PDR). 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 V&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 V&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 V&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 V&V team. Code reviews shall ensure that the source code conforms to the software coding standards and guidelines described in Section 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 IMPLEMENTATION 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 does this review as part of the review process for this SPM.

4.6.2.5 Functional Review

After the design team has completed the test phase, the V&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, PDR, CDR, and if applicable, interim V&V reports (for Protection and Important-to-Safety class software).

Any findings in the functional review shall be documented in the final V&V report.

4.6.2.6 Physical Review

The Physical Review is held to verify that the software and its documentation are internally consistent and are ready for delivery. This shall include a review of the following deliverable items:

- Deliverable software media (in conformance with customer requirements)
- V&V Report and code certificate (for Protection and Important-to-Safety class software)
- User/Technical Manual
- Installation instructions (if required)

The results of physical reviews shall be documented with a Certificate of Conformance in accordance with the requirements of Reference 4.

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. Software inspections may be included as part of the in-process audit activity. The Quality organization shall perform in-process audits for Common Q systems including all software classes. 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
- 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 and prepare plans and schedules for resolving the deficiencies.

4.6.2.8 Managerial Reviews

As part of the Quality organization responsibility, it shall perform the independent managerial 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 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. The review shall be performed by the V&V team, and results documented to identify all deficiencies found. The design team shall plan for the resolution of deficiencies.

4.6.2.10 Post Mortem Review

The EPM shall conduct a project closeout review upon completion of the project to ensure 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 via EXHIBIT 9-2 CORRECTIVE ACTIONS PROCESS. 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 Tests)
- Integration Tests (Pre-FAT)
- System Tests (FAT)
- Installation Tests (SAT)

4.8 PROBLEM REPORTING AND CORRECTIVE ACTION

4.8.1 Purpose and Scope

The purpose of a formal procedure of software problem reporting and corrective action is to ensure 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. V&V activities are the primary vehicle to uncover software problems, while the SCMP shall ensure that actions taken to correct problems by changing configured software are consistent and traceable.

Problem 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 9 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 ensure that the software is 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.

- Automation Engineering management sign-off and approval of all design and V&V documentation shall include one of the following:

  — The ELM of the author, or
  — The EPM

- All members of the Common Q design and V&V team shall be trained in the contents of this SPM. This training shall be documented in the individuals training record.

- 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 Section 6.3.1. Removable storage media should not be switched, renamed, or initialized without prior approval from the EPM, or designee.

A cabinet and disk file cabinet shall be provided for storage and easy access.

4.11.2 Archival Requirements

A locked storage facility shall be used to store all project software in a location separate from the software development area. This locked storage facility shall be able to accommodate the storage of all utilized types of media.

After important Common Q software development milestones or baseline configurations are archived, a known software configuration shall be completely backed up and periodically stored in the vault.

The requirements in this section are the responsibility of the software librarian and should be performed in accordance with Reference 4.

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 RRAS.

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 V&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 ensure that the modifications do not produce unintended adverse effects, and to ensure that the modified software still meets the original software requirements.

Existing software that is not modified shall be qualified for use according to Section 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,
- Problem 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.

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.

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 V&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.

This Page Intentionally Left Blank

(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 V&V process to be applied to Common Q systems. It also defines when, how and by whom specific V&V activities are to be performed including options and alternatives, as required. The section includes various V&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 V&V processes starting with the system design document stage and all necessary V&V activities to verify and/or validate I&C systems. This SVVP complies with Reference 8. The life cycle processes are consistent with References 24 (as augmented by Reference 23).

The goals of this V&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

V&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. V&V activities only include documents and materials identified in SECTION 4.

5.1.2 V&V Program Implementation

V&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 V&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. Many of the tests used by manufacturing are the same or equivalent to those used in the system V&V process. The equivalent of the system validation tests is performed as a minimum on every successive system of the same design that has been previously verified and validated. This is referred to as a Factory Acceptance Test. Traceability of all 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 V&V Documentation

Traceability is important, not only to document the V&V activities, but also to record appropriate actions taken to resolve discrepancies. Thus a V&V program is, by its nature, oriented heavily towards documentation and the ability to trace changes in project documents. All comments generated by the V&V team and all comment resolutions shall be documented consistent with EXHIBIT 4-4 COMMON Q COMMENT RECORD. SECTION 8 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 System Specific V&V Plans

This Common Q V&V plan details the V&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 V&V tasks and resource identification, shall be defined in either a Project Quality Plan or in a project-specific V&V plan that is referenced by a Project Quality Plan, as described in Reference 4.

5.2 REFERENCED DOCUMENTS

Refer to page xxxiii for a list of references.

5.3 DEFINITIONS

Refer to page xxix for a list of acronyms and abbreviations. Refer to page xxxi for definitions.

5.4 VERIFICATION AND VALIDATION OVERVIEW

5.4.1 Organization

An independent V&V team within the Automation Engineering organization performs the safety activities for a given Common Q system implementation project. The V&V team performs software design verification, software validation testing and software configuration status accounting activities on the Common Q systems. The design team may perform the validation testing activities; however, in this case the V&V team shall oversee the conduct of these activities by reviewing documentation and witnessing testing.

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 V&V team is organized independently of the design team. V&V team members may not participate in any design team activities. Also, the V&V team leader, responsible for the V&V, shall be organizationally independent from the design team leader. EXHIBIT 2-1 DESIGN/V&V TEAM ORGANIZATION shows the relationship between the design team and the V&V team. The V&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 V&V schedule and required milestone delivery dates. This shall be developed in coordination with the V&V team leader (for a V&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 ensure 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 to ensure that the platform development meets 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 Verification and Validation Team

V&V team members organizationally report to an Engineering Line Manager (ELM) who is administratively and financially independent from the design team manager. The V&V team ELM provides resource management of people and other resources (such as materials and equipment) to ensure independent implementation of V&V tasks. The composition of the V&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 V&V Team Leader

The V&V team leader is responsible for all technical and administrative matters concerning the verification of the system. The V&V team leader is responsible for the development of the verification requirements and validation test procedure documents. It is also the responsibility of the V&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 the V&V team leader. The verifier is also the independent reviewer for 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 is a key element in the V&V process. The librarian, in the execution of that position ensures that a project's software conforms to library standards, retains records of the software modules in use and revision levels, and assures the procedures for software changes are followed.

5.4.4 Responsibilities

5.4.4.1 Verification and Validation Team Responsibilities

The V&V team shall evaluate the software design and test documentation and perform supplemental testing as deemed necessary.

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.

If warranted, supplemental testing shall be performed without evaluating or using designers test database/bed, or test results.

The emphasis shall be placed on full independent analysis of the system requirements and design specifications, supplemental testing and evaluation for the systems requiring the highest reliability.

The test progress shall be monitored and the execution of tests and test results shall be evaluated. Supplemental testing shall be recommended when found necessary.

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 ensure 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 V&V planning process includes the selection of appropriate tools for a given project.

5.4.5.2 V&V Core Activities

The following V&V core activities are applicable to every system (irrespective of the selected verification test strategy).

1. Upon completion of the V&V review of a particular software item, the reviewer will complete and sign the checklist (SECTION 10) 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 V&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 every system as part of both the development and manufacturing processes; details of test bed, validation test procedures and test results will be documented for review and audit purposes in accordance with Reference 14, Sections 7 and 11 respectively.

4. Unit and Module Testing will be performed by the design team or V&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 Reference 14, Sections 4, 7 and 11 respectively.

5. The V&V team will participate in design reviews, audits and configuration management activities.

6. Commercial off-the-shelf (COTS) software used for Protection and Important-to-Safety class software must go through a Commercial Grade Dedication process as described in Reference 3. A Commercial Grade Dedication report is prepared either by the design team or the V&V team. The V&V team shall review the report to determine its applicability and suitability for meeting the system requirements.

7. 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 V&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.

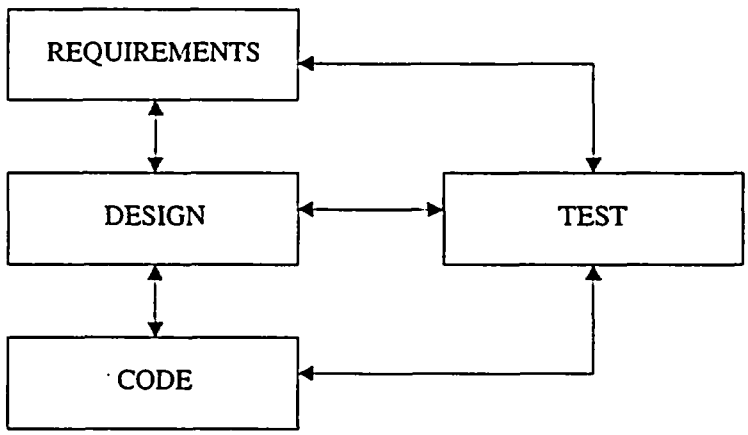
   b. Verify that the changes to the COTS software were performed in accordance with acceptable industry standards (e.g., IEEE 7-4.3.2 or IEC-880). Revise the Commercial Grade Dedication report.

Alternatively, these activities can be performed by the design team and reviewed by the V&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. This shall be the responsibility of the design team. The V&V team shall review the adequacy and accuracy of the RTM. The V&V team shall be responsible for adding information to the RTM related to testing that it performs.

Associating requirements with the documentation and software that satisfy them creates the RTM. The system is tested to show that all requirements have been met. These tests are associated with the requirements on which they are based. 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.



Traceability analysis ensures 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 analysis are organized into successive lower level requirements as described in each document. The purpose of this analysis is to ensure 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 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 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.

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 ensure 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.

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 V&V team:

- data accuracy,
- data completeness,
- data structure, and
- 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 V&V team may also include a review of data accuracy.

Data completeness ensures 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 V&V team should review the database to ensure 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.

V&V database reviews are documented by completing the appropriate sections of EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST.

5.5 LIFE-CYCLE VERIFICATION AND VALIDATION

5.5.1 Management of V&V

The management of V&V spans all life-cycle phases. Software development is a cyclic and iterative process. The V&V effort shall re-perform previous V&V tasks or initiate new V&V tasks to address software changes. V&V tasks are re-performed if errors are discovered in the V&V inputs or outputs.

Management of V&V includes:

1. **Software V&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 V&V activities.

2. **Baseline Change Assessment:** Evaluate proposed software changes for effects on previously completed V&V tasks. When changes are made, plan iteration of affected tasks which includes re-performing previous V&V tasks or initiating new V&V tasks to address the software changes.

3. **Management Review:** Conduct periodic reviews of the V&V process in the area of technical accomplishments, resource utilization, future planning and risk assessment. Support daily management of V&V phase activities. Review final and interim V&V reports. Evaluate V&V results and anomaly resolution to determine when to proceed to the next life-cycle phase and to define changes to V&V tasks to improve the process.

4. **Review Support:** Support management and technical reviews (e.g., Software Requirements Review, Preliminary Design Review, Critical Design Review, etc.). Identify key review support

milestones in SVVP and schedule V&V tasks to meet milestones. Establish methods to exchange V&V data and results with design team.

The costs of V&V shall be identified during the proposal (concept) phase of the project. The resources for performing the V&V shall be identified in the Project Quality Plan (Reference 4) that is prepared by the Project Manager during the conception phase of the software life cycle.

5.5.2 Concept (Initiation) Phase V&V

Concept phase V&V is the period prior to formal definition of the system requirements, which may include a feasibility phase.

Project specific V&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 V&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 V&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 allocation of functions to hardware and software items.
5. Assess criticality of each software item.

5.5.2.3 V&V Outputs

Reporting of the concept review activities can be incorporated in the Requirements Phase report, including identification of deficiencies.

5.5.3 Requirements Phase V&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 selection of these evaluation parameters is based on their ability to assess the quality of the documented communication with the designer and also the user of the system.

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 ensure that the requirements documents do indeed serve the above purpose.

In order to satisfy the need of both the V&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 V&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 V&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. Ensure design requirements are complete, accurate, testable, and unambiguous as possible.

4. Perform software safety requirements analysis.

   a. Identify any hazards and software safety requirements.
   b. Identify any software safety design constraints and guidelines.
   c. Identify any software safety test requirements and provide inputs to the test planning process.
   d. Identify any required, encouraged, discouraged and forbidden design, coding and test techniques.

Verifying the system architecture and decomposition is one of the V&V tasks. The V&V team reviews the interrelationship between hardware/software and subsystems to ensure that the overall integrated system does indeed have potential to meet the system needs and objectives. The following are specific V&V Tasks:

1. Review the adequacy and accuracy of the Requirements Traceability Matrix (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.

2. Perform the following software safety analyses as described in Reference 26, Annex A.1:

   a. Criticality
   b. Specification
   c. Timing and sizing[1]
   d. Different software system (if applicable)

3. Complete EXHIBIT 5-2 CHECKLIST NO. 1, SOFTWARE VERIFICATION AND VALIDATION REQUIREMENTS PHASE CHECKLIST.

4. Other V&V review areas should include:

   a. Review requirements source documents - what is the basis of the requirements?

   b. Review system requirements - is the right problem being solved, are they consistent with emergency procedures, are the right plant parameters being monitored?

   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 - what will be done to see that the operator accepts this system and finds it useful?

5. Review requirements with respect to possible errors. See EXHIBIT 5-2 CHECKLIST NO. 1, SOFTWARE VERIFICATION AND VALIDATION REQUIREMENTS PHASE CHECKLIST for a detailed list of possible errors.

6. Tools used in the development process (such as computers) do not require V&V as long as the resultant code is subject to V&V. Configuration management of these tools will be under the Software Configuration management plan SECTION 6.

[1] This analysis can be performed by the design team and reviewed by the V&V team.

7. Review previously developed or sub-vendor software in the following areas and produce a Commercial Grade Dedication Report stating whether this software is adequate for its intended use. V&V review criteria should be similar to that for newly-developed software or the V&V plan shall state any deviations that are acceptable. Alternatively, these activities can be performed by the design team and reviewed by the V&V team.

   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 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.

The V&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 V&V team in their review.

If the V&V team review of this software finds it acceptable, the V&V team shall insure that the Certificate of Conformance to be issued 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 V&V Outputs

1. Completed EXHIBIT 5-2 CHECKLIST NO. 1, SOFTWARE VERIFICATION AND VALIDATION REQUIREMENTS PHASE CHECKLIST.

2. Produce a report on concept and requirements review activities, including identification of deficiencies.

5.5.4 Design Phase V&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 ensure 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, V&V tasks are conducted on an ongoing basis. This is highly desirable especially when V&V efforts parallel design activities. Test planning and verifying the conformance of design documentation to established standards are the major objectives of preliminary V&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 V&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 V&V Tasks

1. Review system design documentation to ensure 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 ensure 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 the following software safety design analyses as described in Reference 26, Annex A.2

   a. Logic
   b. Data
   c. Interface
   d. Constraint
   e. Functional
   f. Software element

5. Complete EXHIBIT 5-3 CHECKLIST NO. 2, SOFTWARE VERIFICATION AND VALIDATION DESIGN PHASE CHECKLIST.

6. Review the Requirements Traceability Analysis performed by the design team.

5.5.4.3 V&V Outputs

1. Completed EXHIBIT 5-3 CHECKLIST NO. 2, 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.

4. Preparation of a System Test Plan in accordance with Reference 14, Section 3.

5.5.5 Implementation Phase V&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 V&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 V&V Tasks

1. The V&V team shall review the as-built software documentation to ensure the as-built software completely and correctly implements the design specified in the system design documents.

2. Perform the software safety code analyses described in Reference 26, Annex A.3

   a. Logic
   b. Data

c. Interface
d. Constraint
e. Programming style
f. Non-critical code
g. Timing and sizing

3. Review module test reports and unit test reports (if applicable), and verify correct execution of critical software elements.

4. Review the code and associated database(s) for complete and correct implementation of the design. Complete the applicable sections of EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST.

5. Review the hardware/configured software integration procedure to ensure they are complete and correct. Complete the applicable section of EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST.

6. Review the Requirements Traceability Analysis performed by the design team.

7. Evaluate Software Configuration Management activities and ensure the requirements of SECTION 6 are fulfilled.

8. Hardware implementation review is normally conducted as part of the hardware quality assurance activities defined elsewhere.

9. For protection class software, review software testing records to ensure adequate structural testing[1].

10. Begin preparing test procedures in accordance with Reference 14, Section 6. Where special skills are required to prepare test procedures which are only present in design team personnel, then procedures shall be prepared by the design team and reviewed by the V&V team as part of the Test Phase.

5.5.5.3 V&V Outputs

1. Software Module Test.

[1] Structural testing is testing that validates all branches of a software module.

2. Completed EXHIBIT 5-4 CHECKLIST NO. 3, 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.

5.5.6 Test Phase V&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 integrated 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. 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 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 5-5 CHECKLIST NO. 4, 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. Independent supplemental validation tests and scenarios as required to validate the system design.

5. System validation procedures are updated if required.

6. Final developer's documentation, to be:

   a. Complete,
   b. Accurate/compatible with delivered system, and
   c. Compliant with standards.

7. 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 V&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 V&V Tasks

1. Verify program integration with the deliverable hardware per EXHIBIT 5-5 CHECKLIST NO. 4, SOFTWARE VERIFICATION AND VALIDATION TEST PHASE CHECKLIST to insure that all aspects have been considered.

2. System validation test procedures shall be prepared in accordance with Reference 14, Section 6, based upon the requirements of the design and shall include test cases encompassing the range of usage intended for the system. Test procedure(s) shall be either prepared by the design team and reviewed by the V&V team, or prepared by the V&V team and independently reviewed by the design team. 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. These procedures and error correction shall be independently verified in accordance with this V&V plan.

   h. If the validation test procedure is prepared by the design group, the V&V team shall review to address the following:

      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?

3. Perform validation testing in accordance with approved test procedures.

4. The system validation test(s) shall be documented in a report. The report can consist of a completed copy of the test procedure with all blank information completed. The V&V team shall review the report to insure that it includes 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 V&V'd in parallel with change control procedures found in SECTION 6.

   The system validation test report(s) shall also be reviewed by the V&V team to ensure that the following is addressed:

   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 5-5 CHECKLIST NO. 4, SOFTWARE VERIFICATION AND VALIDATION TEST PHASE CHECKLIST.

5. Follow-up on changes and corrections made in the system in accordance with change control procedures in SECTION 6.

6. Review the Requirements Traceability Analysis performed by the design team.

5.5.6.3 V&V Outputs

1. Test procedures.

2. Test Report and evaluation for acceptability.

3. Completed EXHIBIT 5-5 CHECKLIST NO. 4, SOFTWARE VERIFICATION AND VALIDATION TEST PHASE CHECKLIST.

4. Produce a summary report on test phase V&V activity results, including identification of deficiencies and possible enhancements.

5. Code certificates certifying that the software is acceptable for use.

5.5.7 Installation and Checkout Phase V&V

If within Westinghouse's scope of supply, the system installation package shall be reviewed to ensure that all elements necessary to install and operate the system have been correctly and completely specified.

5.5.7.1 V&V Inputs

1. Installation procedures, system generation procedures, etc.
2. User documentation
3. Requirements Traceability Matrix

5.5.7.2 V&V Tasks

1. Review installation procedures and user manuals to ensure they are complete and correct.
2. Review training materials (if within Westinghouse 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. Prepare and issue the final V&V report which provides:

    a. A listing of all V&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 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 V&V.

d. Comments and recommendations to aid in future system upgrades and development.

4. Complete EXHIBIT 5-6 CHECKLIST NO. 5, SOFTWARE VERIFICATION AND VALIDATION INSTALLATION AND CHECKOUT PHASE CHECKLIST.

5. Review the Requirements Traceability Analysis performed by the design team.

5.5.7.3 V&V Outputs

1. Final V&V report with summary review of the system's acceptability.

2. Completed EXHIBIT 5-6 CHECKLIST NO. 5, SOFTWARE VERIFICATION AND VALIDATION INSTALLATION AND CHECKOUT PHASE CHECKLIST.

5.5.8 Operation and Maintenance Phase V&V

Situations may arise after installation of a V&V'd computer system, which may require the performance of additional V&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 V&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 V&V (for example, a code error might not affect the system requirements or design documentation), these areas of V&V may be omitted.

There are two acceptable methods to document the successful completion of V&V activities for program modifications. The first method is to create a V&V report to document all V&V activities regarding the modification. This must include, or reference, a regression analysis including test requirements and results. A new code certificate must be included in the V&V report.

Alternatively, the regression analysis including test requirements and results can be prepared as a separate document. A new code certificate must be prepared that references the regression analysis and the original V&V report.

5.6 SOFTWARE VERIFICATION AND VALIDATION REPORTING

V&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. V&V phase summary reports: These reports are issued after each life cycle phase of the V&V task to summarize the V&V review. Phase summary reports may be consolidated into a single report if desired. These reports shall contain the following:

    a. Description of V&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 4-4 COMMON Q COMMENT RECORD. These reports shall document each discrepancy found during the V&V reviews and include:

    a. Title, number, and revision of document reviewed.
    b. Section/Page reference location.
    c. V&V comment.
    d. Resolution with design team.

3. Final V&V Report: This report shall be issued at the end of the V&V task to summarize and document the V&V activities performed throughout all life cycle phases. The report shall include:

    a. Summary of life cycle V&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 V&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 V&V team must document the resolution in the V&V phase summary reports as well as the final V&V report.

5.7.2 Task Iteration Policy

If the V&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 V&V task. This information should be documented in a Revision Abstract for revised V&V reports, unless a separate regression analysis document is issued in lieu of a revised V&V report.

5.7.3 Deviation Policy

If any deviation is planned from the reviewed and approved V&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 V&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 V&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 V&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 V&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 Reference 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 requirements to be tested, the testing tasks, and the required resources to perform these tasks.

5.8.2 Test Procedure

The test procedure contains the following components:

- Test-Design Specification
- Test-Case Specification
- Test-Procedure Specification

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.

5.9 SOFTWARE INTEGRITY LEVEL SCHEME

The software integrity level refers to 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 Nuclear Classification Assignment 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 1012-1998 is as follows:

Table 5.9-1. Software Classification Mapping

SPM Classification	IEEE Std 1012-1998
Protection	High
Important-To-Safety	Major
Important-To-Availability	Moderate
General Purpose	Low

This Page Intentionally Left Blank

(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. These guidelines meet the intent of U.S. NRC Regulatory Guide 1.169, Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Plants, September, 1997, 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 problem 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 problem reports and changes.

9. Document criteria for generation of software to release for use.

10. Ensure that existing and prior revisions of software can 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 xxix for a list of acronyms and abbreviations. Refer to page xxxi for definitions.

6.1.4 References

Refer to page xxxiii 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 Automation Engineering organization. V&V activities related to configuration management are performed by member(s) of the V&V team.

6.2.2 SCM Responsibilities

The V&V Group in the Automation Engineering organization is 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. Define software items that are to be controlled via this SCMP.

2. Place requirements documentation under configuration control before submittal to the V&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 V&V team for review. Design documentation includes the Software Design Description (SDD).

6.2.2.3 Implementation Phase

1. Place test plans under configuration control.

2. 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 V&V team shall control the test system hardware/software configuration.

3. Place module test procedures and module test reports under configuration control.

4. 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 test 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 V&V report.

7. Place user documentation under configuration control before submittal to the V&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.

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. Ensure 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 EXHIBIT 9-1 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 V&V group ELM, or designee, is responsible for management of SCM activities.

6.2.2.10 Configuration Status Accounting Management

The V&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.

6.2.3 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.

6.3 SOFTWARE CONFIGURATION MANAGEMENT ACTIVITIES

6.3.1 Configuration Identification

All software 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 - The source file should contain a program header block that includes the following information:

Project ####### Module NNNNNNN-VV-RR

where:

#######	is the System number
NNNNNNN	is the Module name (up to 64 alphanumeric characters beginning with a letter, length depending upon the requirements of the file management system utilized)
VV	is the Version number (2 digit sequential number beginning with 00) for successive versions which implement revised software requirements.
RR	is the Revision number (2 digit sequential number beginning with 00 for successive versions which correct errors in the code and require no changes to the software requirements)

The header block should contain a complete revision history of the software item, including comments on each version and revision. In addition, the header block should contain the following information:

- 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:

Project 2000000 Module CALCBLOC-00-01

Control Algorithm Calculation Subroutine

Copyright notice

Description: This program calculates control algorithm setpoint offset values from entered user input of setpoints.

Revision History:

Author:	Date:	Comments:
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.
H. Kim	14-Dec-93	Baseline Version

4. Media, (The physical item containing software items) shall be labeled according to the following conventions:

   Name
   #######-VV-RR ZZZZ XXX of YYY

   where:

Name	A four to 64 alphanumeric name of the software configuration item
#######	is the 7 digit project number
VV	is the two digit successive version number of the Product containing the software
RR	is the Revision number (2 digit sequential number beginning with 00 for successive versions which correct errors in the code and require no changes to the software requirements)
ZZZZ	is the media type (DISK, TAPE, PROM, etc.) which may be more than four characters but not greater than eight characters
XXX	Successive Media item number, beginning with 1
YYY	Total media items of one media type

   An example is:

   CCNPP PAMS RELEASE 0.01
   2000000-00-01 DISK 1 of 2

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 Factory Acceptance Test Report and is sent to the end user (with the Code Certificate and V&V Report) upon delivery of the product. 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

Table 6.3.1-1. Media Identification Example

Media Identification	Software Item Identification
3.5" diskette labeled: 2000000-00-01 DISK 1 of 2	CONTROL1.exe / 01-05 CONTROL1.dat / 00-03 CONTROL1.src / 01-05
3.5" diskette labeled: 2000000-00-01 DISK 2 of 2	CONTROL2.exe / 01-05 CONTROL2.dat / 00-03 CONTROL2.src / 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).

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 location 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 V&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.

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 system baseline has been established 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

SCRs may be initiated by an Exception Report, or by a request for enhancement.

Step 2: Software Change Request Approval/Rejection

The SCR is routed to the following individuals for approval/rejection.

1. EPM
2. Lead software engineer
3. V&V team leader

Each individual determines the feasibility and appropriateness of the change and signs the form for approval/rejection. Rejections must include an explanation for the rejection. Customer/User requests for changes must be approved by the PM or Program Manager.

Step 3: 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.

Step 4: Revised System Baseline

The SCR forms will be used as the basis to track all system changes and to verify 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 V&V 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 reports. These reports when produced shall document the system status at any given time and be maintained by the V&V Group ELM, or designee, for inspection by the customer/user and any auditors.

6.3.4 Configuration Audits And Reviews

1. V&V reviews shall be performed in accordance with this Software Program Manual V&V procedures or a project specific V&V plan.

2. Project 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 the EPM 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. These shall be documented in an audit report.

5. A functional review shall be performed in accordance with Section 4.6.2.5 by the V&V team prior to shipment to verify the software configuration items actual functionality and performance is consistent with the Software Requirements Specification.

6. A physical review shall be performed in accordance with Section 4.6.2.6 by the design team 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.

7. Quality audits may be held at any time by the Quality organization to ensure the software development guidelines, including configuration control, Verification and Validation, and Software Quality Assurance are being adequately executed.

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 ensure that all interfaces external to the Common Q System is effectively controlled:

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 V&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 RRAS shall be documented. Interface between the EPM and the independent V&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 V&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 RRAS, 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 V&V team to be equivalent to this SCMP.

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 V&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 V&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 the V&V guidelines documented in SECTION 5. The evaluation is documented in a Commercial Grade Dedication Report.

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.

6.5 SCM RESOURCES

The V&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 V&V 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 V&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 V&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 V&V team ELM and the Quality Assurance department. The revised plan shall be distributed to all Common Q EPMs doing software design work.

This Page Intentionally Left Blank

(Last Page of Section 6)

SECTION 7 SOFTWARE OPERATION AND MAINTENANCE PLAN

7.1 INTRODUCTION

To specify the requirements for the maintenance and use of Protection class and Important-to-Safety class software used in Common Q Systems. This plan uses Reference 15 as a guide.

Normally, the Platform Lead 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 a Platform Lead 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.

7.2 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.

7.2.1 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 Section 6.3.2.

7.2.2 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 lead engineer for technical analysis and the Platform Lead for approval. They can accept/reject the SCR or request further clarification. If the SCR is approved, then the modification is scheduled by the Platform Lead.

7.2.3 Control

An SCR log shall be maintained for the specific Common Q system implementation. The Platform Lead shall ensure that the approved SCR is entered into this log.

7.2.4 Output

The approved SCR is the output to this process. The original problem report shall be attached to the SCR if applicable. The Platform Lead should be provided an estimate for the modification as input into the next phase.

7.3 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.

7.3.1 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

7.3.2 Analysis Process

This section specifies the process requirements for analyzing the modification.

7.3.2.1 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.

7.3.2.2 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 Reference 14, Section 3 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 software release record.

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 Section 7.3.2.1).

7.3.3 Analysis Control

At this phase of the analysis, the V&V team shall review any changes to the requirements specifications and review the test plan(s) as defined in Sections 5.5.4 and 5.5.8.

The relevant version of project and system documentation from the appropriate configuration control organization (Automation Engineering 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 Platform Lead 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.

7.3.4 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. V&V Requirements Phase Report including RTM

7.4 DESIGN

This section defines the design requirements for software maintenance. Metrics for this phase are defined in Section 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.

7.4.1 Design Input

All outputs from the identification and analysis phases are used as inputs into this phase of software maintenance.

7.4.2 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 Reference 14, Section 6 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.

7.4.3 Design Control

The V&V team shall review the revised SDD as defined in Sections 5.5.4 and 5.5.8 and the test procedures for the modification as defined in Sections 5.5.6 and 5.5.8.

7.4.4 Design Output

The output of the design phase of software maintenance shall include:

1. Revised SDD
2. Test Procedures
3. Design Phase V&V Report including Requirements Traceability Matrix

7.5 IMPLEMENTATION

This section defines the requirements for the implementation phase of software maintenance. Metrics for this phase are defined in Section 4.5.2.4.

7.5.1 Implementation Input

The inputs to the implementation phase shall include all outputs from the identification, analysis and design phases (if applicable).

7.5.2 Implementation Process

The implementation phase shall include the following sub processes.

7.5.2.1 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 V&V activities related to module testing for protection class software is performed in accordance with Sections 5.5.6 and 5.5.8.

7.5.2.2 Integration

Integration is the process of running the revised software in an integrated system environment. It includes informal integration and regression testing to ensure 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.

7.5.2.3 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 V&V team for review per Section 5.5.5.

7.5.2.4 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.

7.5.3 Implementation Control

The V&V activities associated with the implementation phase of the software life cycle as defined in Sections 5.5.5 and 5.5.8 shall be performed to ensure implementation control. The V&V team ELM shall ensure that all software is under software configuration management control in accordance with SECTION 6.

7.5.4 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 V&V report

7.6 TEST

At this phase, formal testing is performed on the new software system.

7.6.1 Test Input

All outputs from the previous phases are used as inputs into this phase of software maintenance.

7.6.2 Test Process

During this phase the V&V team revises or develops new validation test procedures with test cases (if required) to test the modification in accordance with Reference 14, Section 6. If the design team developed the test procedures, then the V&V team reviews the test procedures per Section 5.5.6

After the test procedures have been V&V'd, 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 Section 5.5.6

7.6.3 Test Control

Validation tests shall be conducted, or reviewed, by the V&V team. Any test exceptions shall be documented using the Exception Report form (EXHIBIT 9-1 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 Section 5.5.6.

7.6.4 Test Output

The outputs for the validation test phase of software maintenance are the same as the test phase V&V outputs specified in Section 5.5.6.

7.7 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 Section 4.5.2.4.

7.7.1 Input

The inputs to this phase of software maintenance include the outputs from all previous phases.

7.7.2 Process

A physical configuration audit on the new software system shall be performed according to Section 4.6.2.6. The users of the software shall be notified in accordance with SECTION 9. An archival version of the software shall be performed accordance with SECTION 6.

7.7.3 Control

In addition to the physical audit, the V&V team shall perform the activities associated with the Installation and Checkout Phase, Section 5.5.7.

7.7.4 Output

In addition to the modified software, the outputs for the delivery phase of software maintenance include a final V&V report and Code Certificate.

This Page Intentionally Left Blank

(Last Page of Section 7)

SECTION 8 DOCUMENTATION

8.1 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 to perform a software requirements traceability analysis (RTA) and to maintain a requirements traceability matrix (RTM) as described in Section 5.4.5.3. The author's signature on a document shall signify that the analysis has been completed and the RTM has been updated to reflect the design information contained in the document. All documentation shall comply with Reference 4.

8.2 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.

8.2.1 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.

8.2.2 Software Requirements Specification (SRS)

The Software Requirements Specification shall be developed using Reference 6 and Reference 22 as guidelines. 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.

8.3 SOFTWARE DESIGN DESCRIPTION (SDD)

The software design description shall be prepared using Reference 7 as a guideline.

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.

8.4 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 Section 4.5.2.1.

8.5 SOFTWARE VERIFICATION AND VALIDATION DOCUMENTATION

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

8.5.1 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 V&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 V&V documentation.

For custom software to be developed, the project specifics for V&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 V&V process for the various software categories described in Section 4.1.1
- Software V&V process for existing software to be modified and to be used "as-is".
- Software V&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.

8.5.2 Software Verification and Validation Report

A V&V file shall be maintained by the V&V Team Leader throughout the software life cycle to document all V&V activities. It shall be an ongoing compilation of all validation test results, problem reports and corrective actions (SECTION 9), verification review results and phase reports (Section 4.6), and the results of all Quality audits (Section 4.6.2.7). This file 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 V&V team in accordance with Section 5.5.7.

8.6 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 RRAS Automation Engineering.

8.7 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.

8.8 TEST DOCUMENTATION

This section describes the requirements for test plans and test procedures.

8.8.1 Test Plans

The requirements for test plans can be found in Section 4.3.2.2.

8.8.2 Test Procedures

The requirements for Common Q module, unit, integration and system test procedures can be found in Section 5.5.6.

8.9 COMPUTER CODE CERTIFICATE

The completion of the Implementation and Checkout Phase Software Verification and Validation report is the basis for the issuance of a Computer Code Certificate (EXHIBIT 8-1 COMPUTER CODE CERTIFICATE).

Computer Code Certificates 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.

This Page Intentionally Left Blank

(Last Page of Section 8)

SECTION 9 PROBLEM REPORTING AND CORRECTIVE ACTION

9.1 INTRODUCTION

There are two modes of problem reporting. The first is during the software development phase when validation testing is being performed and test exceptions are found. Section 9.2 describes the reporting process for these errors.

The second mode of error reporting occurs when a user discovers an error after a software release. Section 9.3 describes this reporting process.

Errors shall be documented by completing a form consistent with EXHIBIT 9-1 EXCEPTION REPORT. The exhibit represents the minimum information required. The exception reporting procedure shall be implemented via an automated process.

9.2 ERROR REPORTING BEFORE SOFTWARE RELEASE

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 V&V test process. The following table illustrates the type of report required by each method:

Table 9.2-1. Error Reporting Methods

Method	Report
Verification Reviews	EXHIBIT 4-4 COMMON Q COMMENT RECORD
Validation Tests	EXHIBIT 9-1 EXCEPTION REPORT
General Findings	EXHIBIT 9-1 EXCEPTION REPORT

The appropriate configuration identification data (see Section 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. Copies of all completed Exception Report forms shall be distributed by the independent reviewer to the EPM and the V&V Team ELM and placed under configuration management. 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 Design Group 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 ensure that new errors were not introduced.

9.3 ERROR REPORTING AFTER SOFTWARE RELEASE

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 by completing EXHIBIT 9-1 EXCEPTION REPORT. The Platform Lead shall report errors to all users by issuing Technical Bulletins in accordance with NA Policy 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 considerations.

9.4 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 by completing a formal Issue Report (EXHIBIT 9-2 CORRECTIVE ACTIONS PROCESS).

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 9.1 and 9.2).
- Corrections or changes shall be controlled in accordance with the SCMP (Section 6.3.2).
- Preventive actions and corrective actions are documented on the appropriate form and distributed to the to the appropriate RRAS groups.

Corrective actions shall be documented on Exception Reports and Common Q Comment Records by the design team and shall be completed by the due date specified on the form. If a resolution is not received by the specified due date, the problem shall be escalated to the next level of management. This escalation process is continued through each higher level of management until a response is received. Once the independent reviewer is satisfied with the corrective action taken, the report form shall be signed.

This Page Intentionally Left Blank

(Last Page of Section 9)

SECTION 10 EXHIBITS

This Section contains the following Exhibits:

EXHIBIT 1-1 RELATIONSHIP OF SPM TO IEEE STANDARDS

EXHIBIT 2-1 DESIGN/V&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 4-4 COMMON Q COMMENT RECORD

EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES

EXHIBIT 5-2 CHECKLIST NO. 1, SOFTWARE VERIFICATION AND VALIDATION REQUIREMENTS PHASE CHECKLIST

EXHIBIT 5-3 CHECKLIST NO. 2, SOFTWARE VERIFICATION AND VALIDATION DESIGN PHASE CHECKLIST

EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST

EXHIBIT 5-5 CHECKLIST NO. 4, SOFTWARE VERIFICATION AND VALIDATION TEST PHASE CHECKLIST

EXHIBIT 5-6 CHECKLIST NO. 5, SOFTWARE VERIFICATION AND VALIDATION INSTALLATION AND CHECKOUT PHASE CHECKLIST

EXHIBIT 6-1 SOFTWARE CHANGE REQUEST FORM

EXHIBIT 8-1 COMPUTER CODE CERTIFICATE

EXHIBIT 9-1 EXCEPTION REPORT

EXHIBIT 9-2 CORRECTIVE ACTIONS PROCESS

EXHIBIT 1-1 RELATIONSHIP OF SPM TO IEEE STANDARDS

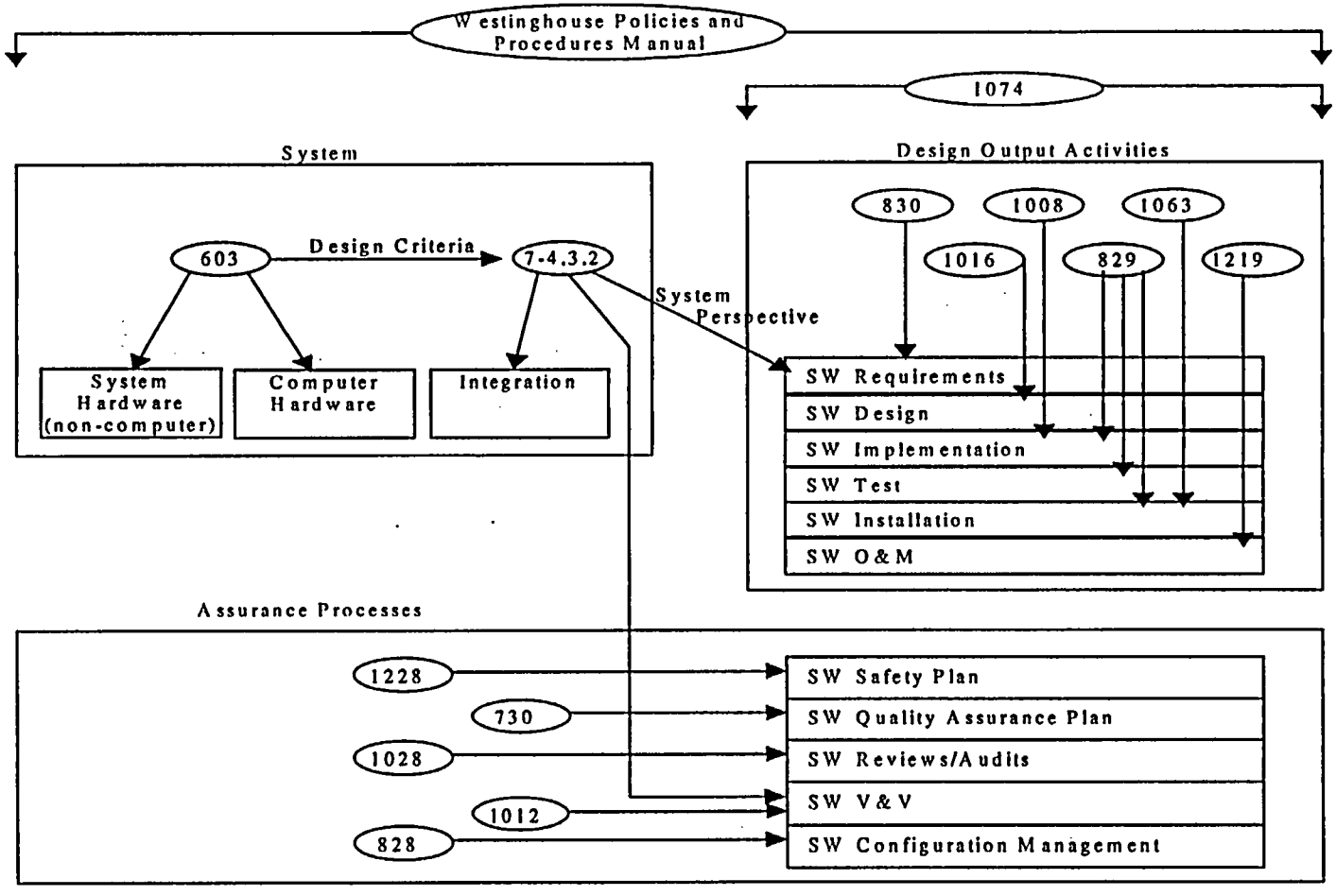


EXHIBIT 2-1 DESIGN/V&V TEAM ORGANIZATION

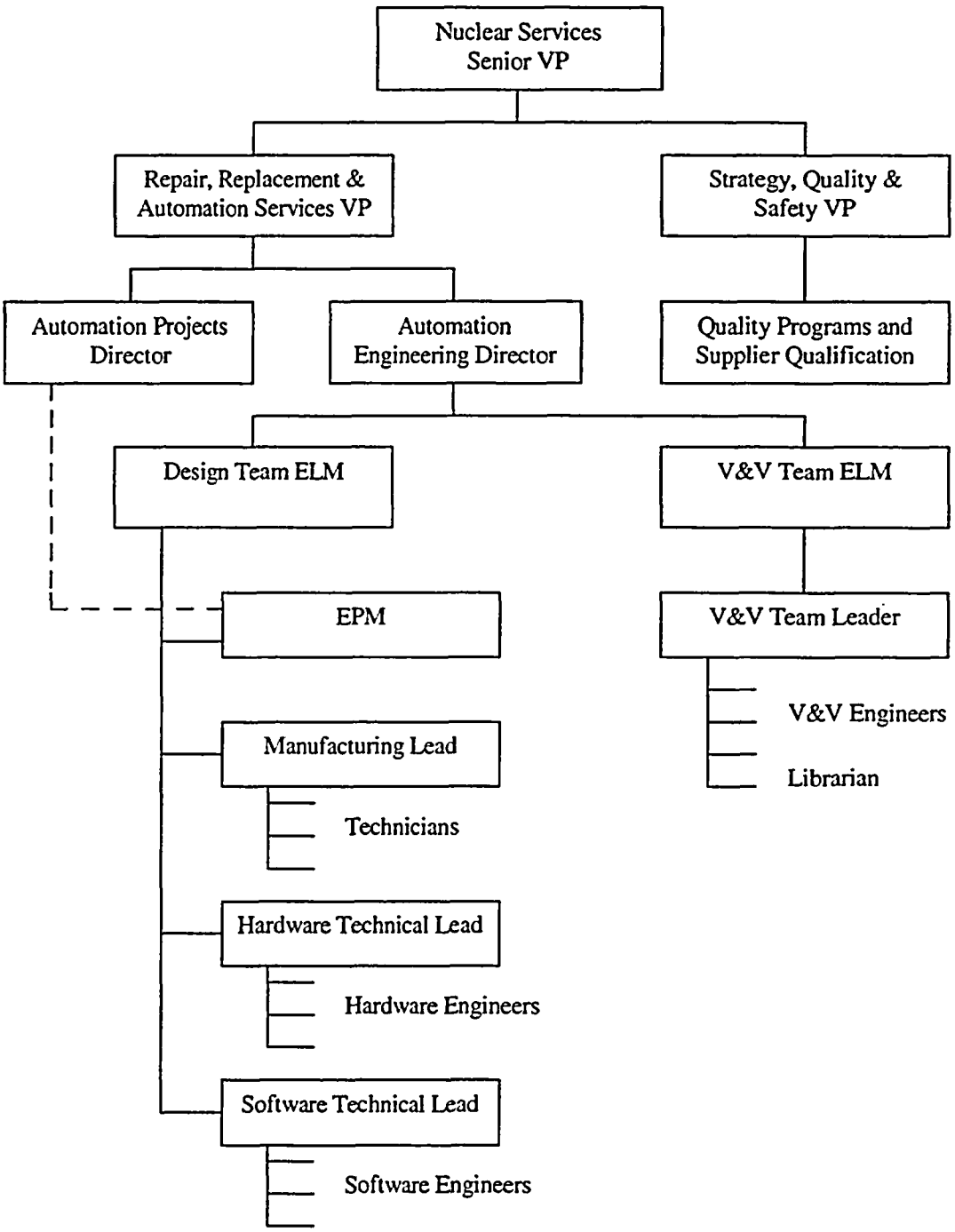


EXHIBIT 4-1 ASSIGNMENT OF COMMON Q SOFTWARE TO CLASSES

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
	Inter-System Communication Software	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
	Inter-System Communication Software	Important-To-Safety
	All Other Software	General Purpose
	Development Tools	General Purpose

EXHIBIT 4-1 ASSIGNMENT OF COMMON Q SOFTWARE TO CLASSES (Continued)

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
	Inter-System Communication Software	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
	Inter-System Communication Software	Important-To-Safety
	All Other Software	General Purpose
	Development Tools	General Purpose

EXHIBIT 4-2 COMMON Q SOFTWARE DEVELOPMENT PROCESS

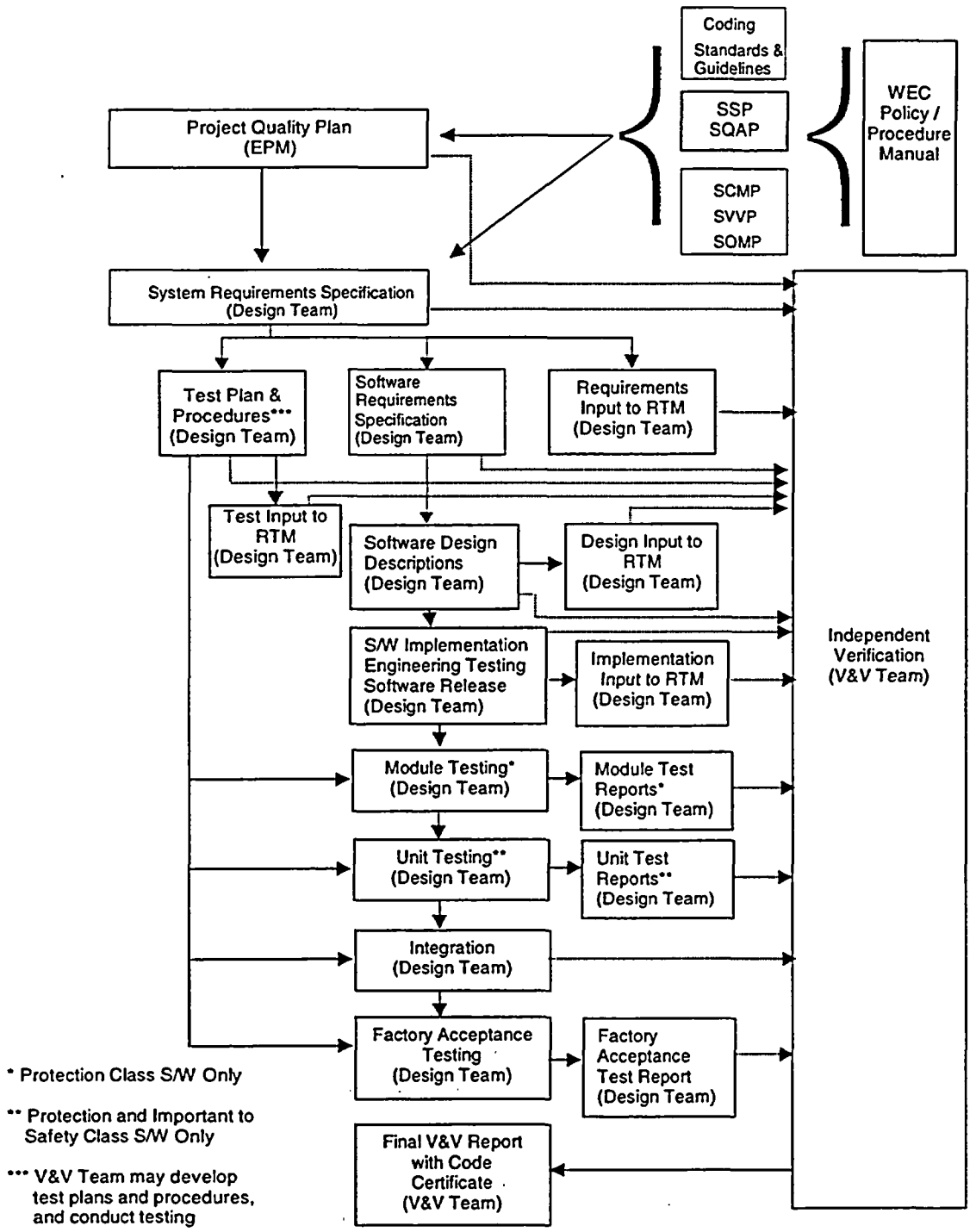


EXHIBIT 4-3 TASKS REQUIRED FOR SOFTWARE CATEGORIES

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 (Continued)

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 TEST (FAT) PROCEDURE	X	X	X
SYSTEM TEST (FAT) EXECUTION	X	X	X
SYSTEM TEST (FAT) REPORT	X	X	X
USER DOCUMENTATION	X	X	X
SOFTWARE V&V REPORT	X	X	X
SOFTWARE INSTALLATION & CHECKOUT PHASE			
INSTALLATION TEST (SAT) PROCEDURE*	X	X	X
INSTALLATION TEST (SAT) EXECUTION*	X	X	X

EXHIBIT 4-3 TASKS REQUIRED FOR SOFTWARE CATEGORIES (Continued)

TASK	ORIGINAL SOFTWARE	ETBM SOFTWARE	ENM SOFTWARE
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 4-4 COMMON Q COMMENT RECORD

DOCUMENT NUMBER / REVISION:		
SECTION REFERENCE	COMMENT	COMMENT RESOLUTION

EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES

TASK	PROTECTION	IMPORTANT-TO-SAFETY	IMPORTANT-TO-AVAILABILITY	GENERAL
SOFTWARE REQUIREMENTS PHASE				
SYSTEM AND SOFTWARE REQUIREMENTS	DT/VT	DT/VT	DT	DT
REQUIREMENTS VERIFICATION	VT	VT	DT	DT
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/DT*	VT/DT*	DT	DT
MODULE CODING	DT/VT	DT/VT	DT	DT
MODULE TEST PROCEDURE	VT/DT*	N/A	N/A	N/A
MODULE TEST EXECUTION / REPORT	VT/DT*	N/A	N/A	N/A
UNIT TEST PROCEDURE	VT/DT*	VT/DT*	N/A	N/A
UNIT TEST EXECUTION / REPORT	VT/DT*	VT/DT*	N/A	N/A
IMPLEMENTATION VERIFICATION	VT	VT	DT	DT
SOFTWARE TEST PHASE				
INTEGRATION TEST PROCEDURE	VT/DT*	VT/DT*	DT	DT

EXHIBIT 5-1 SOFTWARE TASKS AND RESPONSIBILITIES (Continued)

TASK	PROTECTION	IMPORTANT-TO-SAFETY	IMPORTANT-TO-AVAILABILITY	GENERAL
INTEGRATION TEST EXECUTION	VT*	VT*	DT	DT
INTEGRATION TEST REPORT	VT/DT*	VT/DT*	DT	DT
SYSTEM TEST (FAT) PROCEDURE	VT/DT*	VT/DT*	DT	DT
SYSTEM TEST (FAT) EXECUTION	VT*	VT*	DT	DT
SYSTEM TEST (FAT) REPORT	VT/DT*	VT/DT*	DT	DT
USER DOCUMENTATION	DT/VT	DT/VT	DT	DT
SOFTWARE V&V REPORT	VT	VT	N/A	N/A
SOFTWARE INSTALLATION & CHECKOUT PHASE				
INSTALLATION TEST (SAT) PROCEDURE**	VT/DT*	VT/DT*	DT	DT
INSTALLATION TEST (SAT) EXECUTION**	VT*	VT*	DT	DT
INSTALLATION TEST (SAT) REPORT**	VT/DT*	VT/DT*	DT	DT
KEY: ORIGINATOR/REVIEWER (E.G. DT/VT)	DT = DESIGN TEAM	VT = V&V TEAM		

* RRAS ELMs may choose to have the DT write and execute test plans, procedures, and reports and the VT review them. See pages 2-1, 3-3, 4-5, 5-19, and 5-22.

**Applicable if in Westinghouse scope of supply.

Any single software item may be handled, at the discretion of the EPM, according to a class that is more important to safety. For example, a particular important-to-safety software item may be module tested by the V&V team without re-designating the software item as protection class.

EXHIBIT 5-2 CHECKLIST NO. 1, SOFTWARE VERIFICATION AND VALIDATION REQUIREMENTS PHASE CHECKLIST

Software Item Name:_____Software Item ID:_____

1. 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. Determine if the right problem is being solved and if the requirements are consistent with emergency procedures, etc. ___

   b. Completeness and correctness in system definition and interfaces with other equipment. ___

   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, and design 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? ____

2. Do the 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. ___

3. 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). ___

m. After *critical analysis* all software requirements that have safety implications have been identified. Each requirement in the SRS has been evaluated against the various system hazard analyses to assess its potential for unacceptable risk. Refer to IEEE Std. 1228-1994 Annex for guidance. ___

Reviewer's comments (Optional): _____

_____

_____

_____

Reviewed by: Name_____Signature_____Date_____

EXHIBIT 5-3 CHECKLIST NO. 2, SOFTWARE VERIFICATION AND VALIDATION DESIGN PHASE CHECKLIST

Software Item Name:_____Software Item ID:_____

			YES
1.	Does the available documentation adequately address:		
	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. A Software Safety Design Analysis was performed verifying that the safety-critical portion of the software design correctly implements the safety-critical requirements (after Critical Analysis in requirements phase) and introduces no new hazards. Refer to IEEE Std. 1228-1994 Annex for guidance. ___

Reviewer's comments (Optional): _____

_____

_____

_____

Reviewed by: Name_____Signature_____Date_____

EXHIBIT 5-4 CHECKLIST NO. 3, SOFTWARE VERIFICATION AND VALIDATION IMPLEMENTATION PHASE CHECKLIST

Software Item Name:_____Software Item ID:_____

1. Review the source code with respect to the following: YES

   a. Does the source code conform to specified standards and procedures including Reference 31 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? ___

   i. A Software Safety Code Analysis was performed that analyzes the safety-critical portions of the design for correct implementation including: Logic, Data, Interface, Constraint, Programming Style, Non-critical code, Timing/Sizing analyses. Refer to IEEE Std. 1228-1994 Annex for guidance. ___

2. 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 Section 5.4.5.4.) ___

h. Data completeness. (See Section 5.4.5.4.) ___

i. Maintenance. ___

3. 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? Ensure 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? ___

4. 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? Ensure 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? ___

5. 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. ___

Reviewer's comments (Optional): _____

_____

_____

_____

Reviewed by: Name_____ Signature_____ Date_____

EXHIBIT 5-5 CHECKLIST NO. 4, SOFTWARE VERIFICATION AND VALIDATION TEST PHASE CHECKLIST

Software Item Name:_____ Software Item ID:_____

			YES
1.	Verify program integration with hardware in accordance with the following:		
	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?	___
2.	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?	___
3.	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? ___

Reviewer's comments (Optional): _____

_____

_____

_____

Reviewed by: Name_____Signature_____Date_____

EXHIBIT 5-6 CHECKLIST NO. 5, SOFTWARE VERIFICATION AND VALIDATION INSTALLATION AND CHECKOUT PHASE CHECKLIST

Software Item Name:_____Software Item ID:_____

		YES
1.	Is the user documentation installation package sufficient to install the software on the delivered hardware?	___
2.	Is the user documentation clear, unambiguous, and consistent with system requirements?	___
3.	Does the V&V report have positive findings?	___
4.	Have all discrepancies and V&V findings been resolved to the satisfaction of the V&V team?	___
5.	Are SCM controls in place for the user to report errors?	___
6.	A Software Safety Change Analysis was performed for all changes made to the software. Refer to IEEE Std 1228-1994 Annex for guidance.	___
7.	Training documentation meet Safety Training Requirements	___

Reviewer's comments (Optional): _____

_____

_____

Reviewed by: Name_____Signature_____Date_____

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

_____

V&V Approval/Date

Implementation Completed: (Including Documentation)	Testing Completed:
	Exception Report #: _____
_____	Documentation: _____
Implementation Engineer/Date	Review/Date: _____

EXHIBIT 8-1 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:_____

V&V Team Leader:_____Date:_____

EXHIBIT 9-1 EXCEPTION REPORT

Exception Report Number _____

System Name:	Plant:
Procedure Name:	Procedure Number:
Tester Name:	Rev.:
Summary of Exception:	Date:
Class:	Step:
Resolution:	Responsibility:
Implementation:	
☐ Procedure Correction	☐ Software Change
Implemented By:	Date:
Retested By:	Date:
Reviewed By:	Date:

EXHIBIT 9-2 CORRECTIVE ACTIONS PROCESS

Issue Report

Step 1: Identification of Issue - (required information)

Step 1: Identification of Issue

Created by: Alan W Hill
Created on: 08/14/2002
Phone: 8-286-6592

Originator's Organization
Originator's Business Unit: NABU
Organizational Level 2: Projects
Organizational Level 3: Project Controls and Contract Administration
Originator's Group: Projects - Controls & Contract Admin
Originator's Location: Monroeville PA

Issue Title: Example Title
Issue Description: Exmaple description of the issue is entered here.
QA/QC Controlled? ○ Yes ◉ No By choosing yes, this issue will become a CAR
Issue raised by: ◉ Employee ○ Supplier

Step 2: Identification of Issue - (optional information)

Step 2: Identification of Issue - (optional information)

Suggested Significance Level: ○ High ○ Suggestion for Improvement
◉ Medium ○ Best Practice
○ Watch/Trend

Recommended Resolution:
Background:
Date Issue First Occured:
Affected Customer(s): Utility X
Affected Unit(s): Units A & B
Affected Supplier(s):
Immediate Actions Taken:
Impacted Documents, Systems, Components, etc:
Submitted on behalf of:
Is this issue potentially reportable? ○ Yes ○ No ○ I don't know
Improvement Plan #:
Audit #:

Disposition of Material: ○ Accept as is ○ Repair ○ Rework ○ Scrap

Have non-conforming material or calculations been marked or "Hold Tagged"? ○ Yes ○ No ○ N/A

Step 3: Comments
Step 4: Prepare for Issue Review Meeting
Step 5: Issue Review Meeting Complete
Step 6: Event Codes
Step 7: Causal Analysis
Step 8: Root Cause Coding
Step 9: Summary Table of Commitments
Step 10: Effectiveness Evaluation
Step 11: WorkFlow Information
CAM Administration Section

(Last Page of Section 10)